

Package ‘DFBA’

December 6, 2023

Type Package

Title Distribution-Free Bayesian Analysis

Version 0.1.0

Author Daniel H. Barch and Richard A. Chechile

Maintainer Dan Barch <daniel.barch@tufts.edu>

Description Provides tools for Bayesian distribution-free tests.

License GPL-2

Encoding UTF-8

RoxygenNote 7.2.3

Imports methods, graphics, stats

Suggests knitr, rmarkdown, bookdown, testthat (>= 3.0.0), vdiff

Config/testthat/edition 3

VignetteBuilder knitr

Depends R (>= 2.10)

NeedsCompilation no

R topics documented:

DFBA-package	2
dfba_bayes_vs_t_power	2
dfba_beta_bayes_factor	7
dfba_beta_contrast	10
dfba_beta_descriptive	13
dfba_binomial	15
dfba_bivariate_concordance	19
dfba_gamma	22
dfba_gamma_out-class	24
dfba_mann_whitney	24
dfba_mcnemar	29
dfba_median_test	31
dfba_power_curve	34
dfba_sign_test	38
dfba_sim_data	41
dfba_wilcoxon	45
show,dfba_beta_contrast_out-method	49

show,dfba_beta_descriptive_out-method	49
show,dfba_binomial_out-method	50
show,dfba_bivariate_concordance_out-method	50
show,dfba_gamma_out-method	51
show,dfba_interval_BF_out-method	51
show,dfba_mann_whitney_large_out-method	52
show,dfba_mann_whitney_small_out-method	52
show,dfba_mcnemar_out-method	53
show,dfba_median_test_out-method	53
show,dfba_point_BF_out-method	54
show,dfba_power_curve_out-method	54
show,dfba_sign_test_out-method	55
show,dfba_sim_data_out-method	55
show,dfba_t_power_out-method	56
show,dfba_wilcoxon_large_out-method	56
show,dfba_wilcoxon_small_out-method	57

Index 58

DFBA-package	<i>DFBA: Distribution-Free Bayesian Analysis</i>
--------------	--

Description

Provides tools for Bayesian distribution-free tests.

dfba_bayes_vs_t_power	<i>Simulated Distribution-Free Bayesian Power and t Power</i>
-----------------------	---

Description

The function is a design tool for comparing Bayesian distribution-free power versus frequentist t power for a range of sample sizes. Allows for the stipulation of one of nine probability models for data generation.

Usage

```
dfba_bayes_vs_t_power(
  n_min = 20,
  delta,
  model,
  design,
  effect_crit = 0.95,
  shape1 = 1,
  shape2 = 1,
  samples = 1000,
  a0 = 1,
  b0 = 1,
  block_max = 0,
  hide_progress = FALSE
)
```

Arguments

n_min	Smallest desired value of sample size for power calculations (minimum 20; default is also 20)
delta	Offset amount between the two variates
model	Theoretical probability model for the data. One of "normal", "weibull", "cauchy", "lognormal", "chisquare", "logistic", "exponential", "gumbel", or "pareto".
design	Indicates the data structure. One of "independent" or "paired".
effect_crit	Stipulated value for a significant differences for a t -test ($1 - p$), and the critical probability for the Bayesian alternative hypothesis for a Bayesian distribution-free analysis
shape1	The shape parameter for the condition 1 variate for the distribution indicated by the model input (default is 1)
shape2	The shape parameter for the condition 2 variate for the distribution indicated by the model input (default is 1)
samples	Desired number of Monte Carlo data sets drawn to estimate the power (default is 1000)
a0	The first shape parameter for the prior beta distribution (default is 1). Must be positive and finite.
b0	The second shape parameter for the prior beta distribution (default is 1). Must be positive and finite.
block_max	The maximum size for a block effect (default is 0)
hide_progress	(Optional) If TRUE, hide percent progress while Monte Carlo sampling is running. (default is FALSE).

Details

Researchers need to make experimental-design decisions such as the choice about the sample size per condition and the decision of whether to use a within-block design or an independent-groups design. These planning issues arise regardless if one uses either a frequentist or a Bayesian approach to statistical inference. In the DFBA package, there are a number of functions to help users with these decisions. The `dfba_bayes_vs_t_power()` function produces (a) the Bayesian power estimate from a distribution-free analysis and (b) the corresponding frequentist power from a parametric t -test for a set of 11 sample sizes ranging from `n_min` to `n_min + 50` in steps of 5. These estimates are based on a number of different Monte- Carlo-sampled data sets generated by the `dfba_sim_data()` function.

For each data set, statistical tests are performed. If `design = "paired"`, the frequentist t -test is a one-tailed test on the within-block difference scores to assess the null hypothesis that the population mean for E is greater than the population mean for C; if `design = "independent"`, the frequentist t -test is the one-tailed test to assess if there is a significant difference between the two independent conditions (*i.e.* if the mean for condition 2 is significantly greater than the condition 1 mean). If `design = "paired"`, the Bayesian analysis assesses if the posterior probability for $\phi_w > .5$ from the Bayesian Wilcoxon test is greater than `effect_crit`; if `design = "independent"`, the Bayesian analysis assesses if the posterior probability for $\omega_E > .5$ on a Bayesian Mann-Whitney test is greater than `effect_crit`. The frequentist power is estimated by the proportion of the data sets where a parametric t -test detects a significant effect because the upper-tail t value has a p -value less than $1 - \text{effect_crit}$. The Bayesian power is the proportion of the data sets where a posterior probability for the alternative hypothesis is greater than `effect_crit`. The default value for the `effect_crit` argument is `effect_crit = .95`. The frequentist p -value and the Bayesian posterior probability for the alternative hypothesis are calculated using the `dfba_sim_data()` function.

The arguments for the `dfba_sim_data()` function are passed from the `dfba_bayes_vs_t_power()` function. Besides the sample size `n`, there are eight other arguments that are required by the `dfba_sim_data()` function, which are passed from the `dfba_bayes_vs_t_power()` function:

- `a0`
- `b0`
- `model`
- `design`
- `delta`
- `shape1`
- `shape2`
- `block_max`.

The `a0` and `b0` values are the respective first and second beta shape parameters for the prior distribution needed for the Bayesian distribution-free tests, which are ultimately done by calling either the `dfba_wilcoxon()` function or by the `dfba_mann_whitney()` function.

The `model` argument is one of the following strings:

- `"normal"`
- `"weibull"`
- `"cauchy"`
- `"lognormal"`
- `"chisquare"`
- `"logistic"`
- `"exponential"`
- `"gumbel"`
- `"pareto"`

The `design` argument is either `"independent"` or `"paired"`, and stipulates whether the two sets of scores are either independent or from a common block such as for the case of two scores for the same person (*i.e.*, one in each condition).

The `shape1` and `shape2` arguments are values for the shape parameter for the respective first and second condition, and their meaning depends on the probability model. For `model="normal"`, these parameters are the standard deviations of the two distributions. For `model="weibull"`, the parameters are the Weibull shape parameters. For `model="cauchy"`, the parameters are the scale factors for the Cauchy distributions. For `model="lognormal"`, the shape parameters are the standard deviations for $\log(X)$. For `model="chisquare"`, the parameters are the degrees of freedom (*df*) for the two distributions. For `model="logistic"`, the parameters are the scale factors for the distributions. For `model="exponential"`, the parameters are the rate parameters for the distributions.

For the Gumbel distribution, the E variate is equal to $\text{delta} - \text{shape2} * \log(\log(1/U))$ where U is a random value sampled from the uniform distribution on the interval $[.00001, .99999]$, and the C variate is equal to $-\text{shape1} * \log(\log(1/U))$ where U is another score sampled from the uniform distribution. The `shape1` and `shape2` arguments for `model="gumbel"` are the scale parameters for the distributions. The Pareto model is a distribution designed to account for income distributions as studied by economists (Pareto, 1897). For the Pareto distribution, the cumulative function is equal to $1 - (x_m/x)^\alpha$ where x is greater than x_m (Arnold, 1983). In the E condition, $x_m = 1 + \text{delta}$ and in the C condition $x_m = 1$. The alpha parameter is 1.16 times the shape parameters

shape1 and shape2. Since the default value for each shape parameter is 1, the resulting alpha value of 1.16 is the default value. When $\alpha = 1.16$, the Pareto distribution approximates an income distribution that represents the 80-20 law where 20% of the population receives 80% of the income (Hardy, 2010).

The block_max argument provides for incorporating block effects in the random sampling. The block effect for each score is a separate effect for the block. The block effect B for a score is a random number drawn from a uniform distribution on the interval $[0, \text{block_max}]$. When design = "paired", the same random block effect is added to the score in the first condition, which is the random C value, and it is also added to the corresponding paired value for the E variate. Thus, the pairing research design eliminates the effect of block variation for the assessment of condition differences. When design = "independent", there are different block-effect contributions to the E and C variates, which reduces the discrimination of condition differences because it increases the variability of the difference in the two variates. The user can study the effect of the relative discriminability of detecting an effect of delta by adjusting the value of the block_max argument. The default for block_max is 0, but it can be altered to any non-negative real number.

Value

A list containing the following components:

nsims	The number of Monte Carlo data sets; equal to the value of the samples argument
model	Probability model for the data
design	The design for the data; one of "independent" or "paired"
effect_crit	The criterion probability for considering a posterior probability for the hypothesis that $\delta > 0$ to be a detection; it is also $1 - p_crit$ for a frequentist t -test
deltav	The offset between the variates; equal to the delta argument
a0	The first shape parameter for the beta prior distribution
b0	The second shape parameter for the beta prior distribution
block_max	The maximum size of a block effect; equal to block_max argument
outputdf	A dataframe of possible sample sizes and the corresponding Bayesian and frequentist power values

References

- Arnold, B. C. (1983). Pareto Distribution. Fairland, MD: International Cooperative Publishing House.
- Chechile, R. A. (2017). A Bayesian analysis for the Wilcoxon signed-rank statistic. Communications in Statistics - Theory and Methods, <https://doi.org/10.1080/03610926.2017.1388402>
- Chechile, R. A. (2020). A Bayesian analysis for the Mann-Whitney statistic. Communications in Statistics - Theory and Methods, <https://doi.org/10.1080/03610926.2018.1549247>
- Fishman, G. S. (1996) Monte Carlo: Concepts, Algorithms and Applications. New York: Springer.
- Hardy, M. (2010). Pareto's Law. Mathematical Intelligencer, 32, 38-43.
- Johnson, N. L., Kotz S., and Balakrishnan, N. (1995). Continuous Univariate Distributions, Vol. 1, New York: Wiley.
- Pareto, V. (1897). Cours d'Economie Politique. Vol. 2, Lausanne: F. Rouge.

See Also

[Distributions](#) for details on the parameters of the normal, Weibull, Cauchy, lognormal, chi-squared, logistic, and exponential distributions.

[dfba_wilcoxon](#)

[dfba_mann_whitney](#)

[dfba_sim_data](#) for further details about the data for two conditions that differ in terms of their theoretical mean by an amount delta.

Examples

```
# Note: these examples have long runtimes due to Monte Carlo sampling;
# please feel free to run them in the console.

# Examples for two data sets sampled from standard normal distributions with
# no blocking effect

## Not run:
dfba_bayes_vs_t_power(n_min = 40,
                      delta = .45,
                      model = "normal",
                      design = "paired",
                      hide_progress = FALSE)

dfba_bayes_vs_t_power(n_min = 40,
                      delta = .45,
                      model = "normal",
                      design = "independent",
                      hide_progress = FALSE)

# Examples with Weibull-distributed variates with no blocking effect

dfba_bayes_vs_t_power(n_min = 50,
                      delta = .45,
                      model = "weibull",
                      design = "paired",
                      hide_progress = FALSE)

dfba_bayes_vs_t_power(n_min = 50,
                      delta = .45,
                      model = "weibull",
                      design = "independent",
                      hide_progress = FALSE)

# Examples with Weibull-distributed variates with a blocking effect

dfba_bayes_vs_t_power(n_min = 50,
                      delta = .45,
                      model = "weibull",
                      design = "independent",
                      shape1 = .8,
                      shape2 = .8,
                      block_max = 2.3,
                      hide_progress = FALSE)
```

```
dfba_bayes_vs_t_power(n_min = 50,
                      delta = .45,
                      model = "weibull",
                      design = "paired",
                      shape1 = .8,
                      shape2 = .8,
                      block_max = 2.3,
                      hide_progress = FALSE)

## End(Not run)
```

dfba_beta_bayes_factor

Bayes Factor for Posterior Beta Distribution

Description

Given a beta posterior distribution and given a prior for the variate, computes the Bayes factor for either point or interval null hypotheses.

Usage

```
dfba_beta_bayes_factor(a_post, b_post, method, H0, a0 = 1, b0 = 1)
```

Arguments

a_post	The first shape parameter for the posterior beta distribution. Must be positive and finite.
b_post	The second shape parameter for the posterior beta distribution. Must be positive and finite.
method	One of "interval" if the null hypothesis is a range on the $[0, 1]$ interval or "point" if the null hypothesis is a single number in the $[0, 1]$ interval
H0	If method="interval", then the H0 input is vector of two values, which are lower and upper limits for the null hypothesis; if method="point", then the H0 input is single number, which is the null hypothesis value
a0	The first shape parameter for the prior beta distribution (default is 1). Must be positive and finite.
b0	The second shape parameter for the prior beta distribution(default is 1). Must be positive and finite.

Details

For a binomial variate with n_1 successes and n_2 failures, the Bayesian analysis for the population success rate parameter ϕ is distributed as a beta density function with shape parameters a_post and b_post for a_post = $n_1 + a_0$ and b_post = $n_2 + b_0$ where a_0 and b_0 are the shape parameters for the prior beta distribution. It is common for users to be interested in testing hypotheses about the population ϕ parameter. The Bayes factor is useful to assess if either the null or the alternative hypothesis are credible.

There are two types of null hypotheses – an interval null hypothesis and a point null hypothesis. For example, an interval null hypothesis might be $\phi \leq .5$ with the alternative hypothesis being

$\phi > .5$, whereas a point null hypothesis might be $\phi = .5$ with the alternative being $\phi \neq .5$. It is conventional to call the null hypothesis H_0 and to call the alternative hypothesis H_1 . For frequentist null hypothesis testing, H_0 is assumed to be true, to see if this assumption is likely or not. With the frequentist approach the null hypothesis cannot be proved since it was assumed in the first place. With frequentist statistics, H_0 is thus either retained as assumed or it is rejected. Unlike the frequentist approach, Bayesian hypothesis testing does not assume either H_0 or H_1 ; it instead assumes a prior distribution for the population parameter ϕ , and based on this assumption arrives at a posterior distribution for the parameter given the data of n_1 and n_2 for the binomial outcomes.

There are two related Bayes factors - BF_{10} and BF_{01} where $BF_{01} = 1/BF_{10}$. When $BF_{10} > 1$, there is more support for the alternative hypothesis, whereas when $BF_{01} > 1$, there is more support for the null hypothesis. Thus, in Bayesian hypothesis testing it is possible to build support for either H_0 or H_1 . In essence, the Bayes factor is a measure of the relative strength of evidence. There is no standard guideline for recommending a decision about the prevailing hypothesis, but several statisticians have suggested criteria. Jeffreys (1961) suggested that $BF > 10$ was *strong* and $BF > 100$ was *decisive*; Kass and Raffrey (1995) suggested that $BF > 20$ was *strong* and $BF > 150$ was *decisive*. Chechile (2020) argued from a decision-theory framework for a third option for the user to decide *not to decide* if the prevailing Bayes factor is not sufficiently large. From this decision-making perspective, Chechile (2020) suggested that $BF > 19$ was a *good bet - too good to disregard*, $BF > 99$ was a *strong bet - irresponsible to avoid*, and $BF > 20,001$ was *virtually certain*. Chechile also pointed out that despite the Bayes factor value there is often some probability, however small, for either hypothesis. Ultimately, each academic discipline has to set the standard for their field for the strength of evidence. Yet even when the Bayes factor is below the user's threshold for making claims about the hypotheses, the value of the Bayes factor from one study can be nonetheless valuable to other researchers and might be combined *via* a product rule in a meta-analysis. Thus, the value of the Bayes factor has a descriptive utility.

The Bayes factor BF_{10} for an interval null is the ratio of the posterior odds of H_1 to H_0 divided by the prior odds of H_1 to H_0 . Also, the converse Bayes factor BF_{01} is the ratio of posterior odds of H_0 to H_1 divided by the prior odds of H_0 to H_1 ; hence $BF_{01} = 1/BF_{10}$. If there is no change in the odds ratio as a function of new data being collected, then $BF_{10} = BF_{01} = 1$. But, if evidence is more likely for one of the hypotheses, then either BF_{10} or BF_{01} will be greater than 1.

The population parameter ϕ is distributed on the continuous interval $[0, 1]$. The prior and posterior beta distribution are probability density displays. Importantly, this means that no point has a nonzero probability density, even as the probability mass for any mathematical point is zero. For this reason, all point null hypotheses have a probability measure of zero, but can have a probability density that can be different for prior and posterior distributions. There still is a meaningful Bayes factor for a point hypothesis. As described in Chechile (2020),

$$BF_{10} = [p(H_1|D)/p(H_1)][p(H_0)/p(H_0|D)]$$

where D denotes the data. The first term in this equation is $1/1 = 1$. But the second term is of the form $0/0$, which appears to be undefined. However, by using L'Hospital's rule, it can be proved that the term $p(H_0)/p(H_0|D)$ is the ratio of prior probability density at the null point divided by the posterior probability density. This method for finding the Bayes factor for a point is called the Savage-Dickey method because of the separate contributions from both of those statisticians (Dickey & Lientz, 1970).

Value

A list containing the following components:

method	The string of either "interval" or "point" corresponding to the type of null hypothesis tested
--------	--


```
dfba_beta_bayes_factor(a_post = 273,
                      b_post = 278,
                      method = "interval",
                      H0 = c(.4975,
                           .5025)
                      )
```

dfba_beta_contrast	<i>Bayesian Contrasts</i>
--------------------	---------------------------

Description

This function implements a Bayesian analysis of a linear contrast of conditions when there are 2 or more independent conditions and where the variate for each condition is a binomial.

Usage

```
dfba_beta_contrast(
  n1_vec,
  n2_vec,
  contrast_vec,
  a0_vec = rep(1, length(n1_vec)),
  b0_vec = rep(1, length(n1_vec)),
  prob_interval = 0.95,
  samples = 10000
)
```

Arguments

n1_vec	A vector of length K that consists of the observed number of successes for the categorical variable in each of the K separate conditions
n2_vec	A vector of length K that consists of the observed number of failures for the categorical variable in each of the K separate conditions
contrast_vec	A vector of coefficients of a linear comparison among the conditions where the sum of all the coefficients must be 0 and the sum of the positive coefficients must be 1 and the sum of the negative coefficients must be -1
a0_vec	A vector of length K that consists of the prior a0 shape parameters for the separate betas (the default values are 1)
b0_vec	A vector of length K that consists of the prior b0 shape parameters for the separate betas (the default values are 1)
prob_interval	Desired probability for equal-tail interval estimate on the contrast (default is 0.95)
samples	The desired number of Monte Carlo samples taken from each posterior beta variate (default is 10000)

Details

Since the Bayesian analysis for each separate condition has a posterior beta distribution with known shape parameters, the program approximates, *via* Monte Carlo sampling, a linear contrast among the set of independent beta distributions because the contrast of beta distributions is not a known probability model.

Given a binomial categorical variate for each of K independent conditions with $K \geq 2$, the standard frequentist nonparametric analysis is to do a χ^2 test with $K - 1$ degrees of freedom (Siegel & Castellan, 1988). Hypothesis testing for the frequentist χ^2 test assesses the sharp-null hypothesis that the binomial success rate is exactly equal in all the conditions. But this point-null hypothesis is not an interesting question about the population success rate from a Bayesian viewpoint because the probability of any single point hypothesis has a probability measure value of zero (Chechile, 2020). Although it is possible that the frequentist null hypothesis can be retained for small- n studies, the hypothesis itself is about the population in the case of unlimited sample size, and surely for this limiting case it is almost certain that the hypothesis is not exactly true. Thus, from the Bayesian framework, the point- null hypothesis is not a good use of scientific effort and resources, and it is more scientifically meaningful to assess a linear comparison of the conditions, such as to assess if the population success rate in one condition is greater than the success rate in another condition. An interval hypothesis such as this has a meaningful probability value, as does the complimentary hypothesis. If ϕ_1 and ϕ_2 are, respectively, the population success rates for the binomials in conditions 1 and 2, then a meaningful comparison might be to assess the probability distribution for $\Delta = \phi_2 - \phi_1$. This example is a simple linear contrast with contrast coefficient weights of -1 and 1, which are the multipliers for the two population success rates. If the posterior interval estimate for the contrast contains 0, then the hypothesis of $\Delta = 0$ has some credibility in light of the given sample size. Thus, by estimating the distribution of Δ , the user learns important information about condition differences. As another example of a contrast, suppose there are three conditions where the first condition is a standard control and the other two conditions are different alternative conditions. In this case, a user might want to compare the mean of the control data against the average of the two experimental- condition means, *i.e.*, the contrast of

$$\Delta = -1\phi_1 + .5\phi_2 + .5\phi_3.$$

In this second example, the coefficients of the contrast are $[-1, +.5, +.5]$. As a third example, the user might also be interested in a comparison where the two experimental conditions are compared, *i.e.*, the contrast of

$$\Delta = 0\phi_1 + 1\phi_2 - 1\phi_3.$$

For the `dfba_beta_contrast()` function, the user is required to stipulate the coefficients of a contrast such that the sum of all the coefficients is 0, the sum of the positive coefficients is 1, and the sum of the negative coefficients is -1. This constraint on the coefficients forces Δ to be on the $[-1, +1]$ interval.

There is a standard Bayesian posterior for each condition, which is a beta distribution (see Chechile (2020) for a detailed discussion of this literature). In short, it is well known that the beta distribution is a natural Bayesian conjugate function for Bernoulli random processes. Thus, a prior beta distribution with shape parameters a_0 and b_0 results (*via* Bayes's theorem) in a posterior beta with shape parameters a and b where $a = a_0 + n_1$ and $b = b_0 + n_2$, where n_1 and n_2 are the respective successes and failures of the categorical variable. While the Bayesian analysis of each beta distribution for the separate conditions are known, a comparison among 2 or more separate beta distributions is not distributed as a beta. The posterior mean of a linear contrast of separate beta variates has a known mean regardless of the correlations among the variates, but the distributional form of the contrast of independent betas is not known in closed form. The distributional form is important for ascertaining issues such as determining the probability that the contrast is positive or specifying a probability interval for the contrast. But, with the `dfba_beta_contrast()` function, these important aspects of the Bayesian analysis are approximated via Monte Carlo simulation.

The `samples` argument stipulates the number of random values to be drawn from each of the K posterior conditions. The default value for `samples` is 10000. The default value of 10000 is also the minimum value that can be selected (increased values of `samples` provide increased precision). Posterior interval estimation and the Bayes factor for the contrast are provided on the basis of the Monte Carlo sampling. If `samples` is equal to N and if ϕ_1, \dots, ϕ_K are the parameters for the population success rates, then there are N random values drawn from each of ϕ_i parameters for $i = 1, \dots, K$. Given the contrast coefficients stipulated in the arguments, there are N delta random posterior values where $\Delta_j = \Psi_1\phi_{1j} + \dots + \Psi_i\phi_{Kj}$ for $j = 1, \dots, N$, where Ψ_i are the contrast coefficients specified in the `contrast_vec` argument. The Monte Carlo sampling from each posterior beta with known shape parameters uses the `rbeta()` function. Thus, unlike Bayesian procedures that employ Markov chain Monte Carlo algorithms, the Monte Carlo sampling in the `dfba_beta_contrast()` function does not depend on a burn-in process or a starting estimate. Thus, all the N sampled values are valid random samples. Repeated use of the `dfba_beta_contrast()` function for the same input will naturally exhibit some random variation in the interval estimate and in the Bayes factor for a contrast greater than 0. However, the point estimate for the contrast does not depend on the Monte Carlo sampling, and it is constant given the vectors for `n1_vec` and `n2_vec` and given the same prior.

Value

A list containing the following components:

<code>mean</code>	Exact posterior mean estimate for the contrast
<code>eti_lower</code>	The lower equal-tail limit for the contrast for the probability interval value specified by <code>prob_interval</code>
<code>eti_upper</code>	The upper equal-tail limit for the contrast for the probability interval value specified by <code>prob_interval</code>
<code>prob_positive_delta</code>	Posterior probability that the contrast is positive
<code>prior_positive_delta</code>	Prior probability that the contrast is positive
<code>bayes_factor</code>	The Bayes factor for the posterior-to-prior odds for a positive contrast to a non-positive contrast
<code>delta_quantiles</code>	Quantile values (<code>probs = seq(0, 1, 0.01)</code>) for the posterior contrast from the Monte Carlo sampling
<code>a_vec</code>	A vector of length K that consists of the posterior <code>a</code> shape parameters for the separate posterior beta distributions
<code>b_vec</code>	A vector of length K that consists of the posterior <code>b</code> shape parameters for the separate posterior beta distributions
<code>a0_vec</code>	A vector of length K that consists of the prior <code>a0</code> shape parameters for the separate prior beta distributions
<code>b0_vec</code>	A vector of length K that consists of the prior <code>b0</code> shape parameters for the separate prior beta distributions
<code>contrast_vec</code>	A vector for the contrast coefficients for a linear comparison of posterior beta variates
<code>prob_interval</code>	The probability for the equal-tail estimate for the contrast (default is 0.95)
<code>samples</code>	The number of Monte Carlo samples from the K separate posterior beta distributions

References

- Chechile, R. A. (2020). Bayesian Statistics for Experimental Scientists: A General Introduction Using Distribution-Free Methods. Cambridge: MIT Press.
- Siegel, S. & Castellan, N. J. (1988). Nonparametric Statistics for the Behavioral Sciences. New York: McGraw Hill.

Examples

```
## Suppose there are four conditions from a factorial design
# where the conditions labels are A1B1, A2B1, A1B2, and A2B2
# where the frequencies for success for the binomial variate are:
n1_vec <- c(22, 15, 13, 21)
# and the frequencies for failures per condition are:
n2_vec <- c(18, 25, 27, 19)
# Let us test the following three orthogonal contrasts
contrast.B1vsB2 <- c(.5, .5, -.5, -.5)
contrast.A1vsA2 <- c(.5, -.5, .5, -.5)
contrast.ABinter <- c(.5, -.5, -.5, .5)

dfba_beta_contrast(n1_vec = n1_vec,
                  n2_vec = n2_vec,
                  contrast_vec = contrast.B1vsB2)

dfba_beta_contrast(n1_vec,
                  n2_vec,
                  contrast_vec = contrast.A1vsA2)

dfba_beta_contrast(n1_vec,
                  n2_vec,
                  contrast_vec = contrast.ABinter)

# Plot the cumulative distribution for AB interaction
testABinteraction<-dfba_beta_contrast(n1_vec,
                                     n2_vec,
                                     contrast_vec = contrast.ABinter)

plot(testABinteraction)
```

dfba_beta_descriptive *Descriptive Statistics for a Beta Distribution*

Description

Given the two shape parameters for a beta distribution, the function provides central tendency statistics, interval limits, and density and cumulative probabilities.

Usage

```
dfba_beta_descriptive(a, b, prob_interval = 0.95)
```

Arguments

- | | |
|---------------|--|
| a | The first shape parameter for the beta distribution. Must be positive and finite. |
| b | The second shape parameter for the beta distribution. Must be positive and finite. |
| prob_interval | Desired probability within interval limits (default is .95) |

Details

The density function for a beta variate is

$$f(x) = \begin{cases} Kx^{a-1}(1-x)^{b-1} & \text{if } 0 \leq x \leq 1, \\ 0 & \text{otherwise} \end{cases}$$

where

$$K = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}.$$

(Johnson, Kotz, & Balakrishnan, 1995). The two shape parameters a and b must be positive values.

The `dfba_beta_descriptive()` function provides features to complement the beta distribution functions available in the **stats** package. The function provides the mean, median, mode, and variance for a beta variate in terms of its two shape parameters.

While the mean, variance, and median are straightforward, there are several conditions that result in an undefined mode. When either (1) $a = b = 1$, (2) $a < 1$, or (3) $b < 1$, the mode is undefined. For example, when $a = b = 1$, the function is the uniform distribution, which does not have a modal value. The other cases above result in the density function diverging at either $x = 0$ or $x = 1$. The function returns a value of NA for the mode for all the cases where a unique mode does not exist.

For interval estimation, the function finds an equal-tail interval limits in all cases, and it also provides the highest-density limits when there is a well-defined mode. When the mode does not exist, the function returns NA for the limits for the highest-density interval (HDI). For interval estimation, the probability between the lower and upper limit is the probability specified in the `prob_interval` input. The `dfba_beta_descriptive()` output object includes a dataframe that has density and cumulative probability information that can be used for plotting.

Value

A list containing the following components:

<code>a</code>	The first beta shape parameter
<code>b</code>	The second beta shape parameter
<code>prob_interval</code>	The probability for interval estimates
<code>x_mean</code>	The mean of the distribution
<code>x_median</code>	The median of the distribution
<code>x_mode</code>	The mode for the distribution
<code>x_variance</code>	The variance for the distribution
<code>eti_lower</code>	The equal-tail lower interval limit
<code>eti_upper</code>	The equal-tail upper interval limit
<code>hdi_lower</code>	The lower limit for the highest-density interval
<code>hdi_upper</code>	The upper limit for the highest-density interval
<code>outputdf</code>	A dataframe of <code>x</code> , density, and cumulative probability for <code>x</code> from 0 to 1 in steps of .005

References

Johnson, N. L., Kotz S., and Balakrishnan, N. (1995). *Continuous Univariate Distributions*, Vol. 1, New York: Wiley.

See Also

[Distributions](#) for additional details on functions for the beta distribution in the **stats** package.

Examples

```
dfba_beta_descriptive(a = 38,
                      b = 55)

dfba_beta_descriptive(38,
                      55,
                      prob_interval=.99)
```

dfba_binomial	<i>Bayesian Binomial Rate Parameter Inference</i>
---------------	---

Description

Given binomial frequency data, provides a Bayesian analysis for the population binomial rate parameter.

Usage

```
dfba_binomial(n1, n2, a0 = 1, b0 = 1, prob_interval = 0.95)
```

Arguments

n1	Integer number of binomial observations for a category 1 response (<i>e.g.</i> , the number of successes)
n2	Integer number of binomial observations for a category 2 response (<i>e.g.</i> , the number of failures)
a0	The first shape parameter for the prior beta distribution that corresponds to the population binomial parameter (default is 1). Must be positive and finite.
b0	The second shape parameter for the prior beta distribution for the population binomial rate parameter (default is 1). Must be positive and finite.
prob_interval	Probability within interval estimates for the population binomial rate parameter (default is .95)

Details

The binomial distribution with size = n and probability = ϕ has discrete probabilities

$$p(x) = \frac{n!}{x!(n-x)!} \phi^x (1-\phi)^{n-x}$$

where x is an integer from 0 to n in steps of 1. The binomial model assumes a Bernoulli process of independent trials where there are binary outcomes that have the same probability (say, ϕ) for a response in one of the two categories and a probability of $1 - \phi$ for the other category. Before any data are collected, there are $n + 1$ possible values for x number of outcomes in category 1 and $n - x$ number of outcomes in category 2. The binomial distribution is a likelihood distribution. A

likelihood is the probability of an outcome given a specific value for the population rate parameter. Yet for real applications, the population parameter is not known. All that is known are the outcomes observed from a set of binomial trials. The binomial inference problem is to estimate the population ϕ parameter based on the sample data.

The frequentist approach to statistics is based on the relative frequency method of assigning probability values (Ellis, 1842). From this framework, there are no probabilities for anything that does not have a relative frequency (von Mises, 1957). In frequency theory, the ϕ parameter does not have a relative frequency, so it cannot have a probability distribution. From a frequentist framework, a value for the binomial rate parameter is *assumed*, and there is a discrete distribution for the $n + 1$ outcomes for x from 0 to n . The discrete likelihood distribution has relative frequency over repeated experiments. Thus, for the frequentist approach, x is a random variable, and ϕ is an unknown fixed constant. Frequency theory thus deliberately eschews the idea of the binomial rate parameter having a probability distribution. Laplace (1774) had previously employed a Bayesian approach of treating the ϕ parameter as a random variable. Yet Ellis and other researchers within the frequentist tradition deliberately rejected the Bayes/Laplace approach. For tests of a null hypothesis of an assumed ϕ value, the frequentist approach either continues to assume the null hypothesis or it rejects the null hypothesis depending on the likelihood of the observed data plus the likelihood of more extreme unobserved outcomes. The confidence interval is the range of ϕ values where the null hypothesis of specific ϕ values would be retained given the observed data (Clopper & Pearson, 1934). However, the frequentist confidence interval is not a probability interval since population parameters cannot have a probability distribution with frequentist methods. Frequentist statisticians were well aware (e.g., Pearson, 1920) that if the ϕ parameter had a distribution, then the Bayes/Laplace approach would be correct.

Bayesian statistics rejects the frequentist theoretical decisions as to what are the fixed constants and what is the random variable that can take on a range of values. From a Bayesian framework, probability is anything that satisfies the Kolmogorov (1933) axioms, so probabilities need not be limited to processes that have a relative frequency. Importantly, probability can be a measure of information or knowledge provided that the probability representation meets the Kolmogorov axioms (De Finetti, 1974). Given binomial data, the population binomial rate parameter ϕ is unknown, so it is represented with a probability distribution for its possible values. This assumed distribution is the prior distribution. Furthermore, the quantity x for the likelihood distribution above is not a random variable once the experiment has been conducted. If there are n_1 outcomes for category 1 and $n_2 = n - n_1$ outcomes in category 2, then these are fixed values. While frequentist methods compute both the likelihood of the observed outcome *and* the likelihood for unobserved outcomes that are more extreme, in Bayesian inference it is *only* the likelihood of the observed outcome that is computed. From the Bayesian perspective, the inclusion of unobserved outcomes in the analysis violates the likelihood principle (Berger & Wolpert, 1988). A number of investigators have found paradoxes with frequentist procedures when the likelihood principle is not used (e.g., Lindley & Phillips, 1976; Chechile, 2020). The Bayesian practice of strictly computing only the likelihood of the observed data produces the result that the likelihood for the binomial is proportional to $\phi^{n_1}(1 - \phi)^{n_2}$. In Bayesian statistics, the proportionality constant is not needed because it appears in both the numerator and the denominator of Bayes theorem and thus cancels. See Chechile (2020) for more extensive comparisons between frequentist and Bayesian approaches with a particular focus on the binomial model.

Given a beta distribution prior for the binomial ϕ parameter, it has been shown that the resulting posterior distribution from Bayes theorem is another member of the beta family of distributions (Lindley & Phillips, 1976). This property of the prior and posterior being in the same distributional family is called *conjugacy*. The beta distribution is a natural Bayesian conjugate function for all Bernoulli processes where the likelihood is proportional to $\phi^{n_1}(1 - \phi)^{n_2}$ (Chechile, 2020). The

density function for a beta variate is

$$f(x) = \begin{cases} Kx^{a-1}(1-x)^{b-1} & \text{if } 0 \leq x \leq 1, \\ 0 & \text{otherwise} \end{cases}$$

where

$$K = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}$$

(Johnson, Kotz, & Balakrishnan, 1995). The two shape parameters a and b must be positive values. If the beta prior shape parameters are a_0 and b_0 , then the posterior beta shape parameters are $a_{post} = a_0 + n_1$ and $b_{post} = b_0 + n_2$. The default prior for the `dfba_binomial()` function is $a_0 = b_0 = 1$, which corresponds to the uniform prior.

Thus, the Bayesian inference for the unknown binomial rate parameter ϕ is the posterior beta distribution with shape parameters of `a_post` and `b_post`. The `dfba_binomial()` function calls the `dfba_beta_descriptive()` function to find the centrality point estimates (*i.e.*, the mean, median, and mode) and to find two interval estimates that contain the probability specified in the `prob_interval` argument. One interval has equal-tail probabilities and the other interval is the highest-density interval. Users can use the `dfba_beta_bayes_factor()` function to test hypotheses about the ϕ parameter.

Value

A list containing the following components:

<code>n1</code>	Observed number of category 1 responses
<code>n2</code>	Observed number of category 2 responses
<code>a0</code>	First shape parameter for the prior beta distribution of the binomial rate parameter
<code>b0</code>	Second shape parameter for the prior beta distribution of the binomial rate parameter
<code>prob_interval</code>	Probability within interval estimates for the population binomial rate parameter
<code>a_post</code>	First shape parameter for the posterior beta distribution for the binomial rate parameter
<code>b_post</code>	Second shape parameter for the posterior beta distribution for the binomial rate parameter
<code>phimean</code>	Mean of the posterior beta distribution for the binomial rate parameter
<code>phimedian</code>	Median of the posterior beta distribution for the binomial rate parameter
<code>phimode</code>	Mode of the posterior beta distribution for the binomial rate parameter
<code>eti_lower</code>	Lower limit for the posterior equal-tail interval that has the probability stipulated in the <code>prob_interval</code> argument
<code>eti_upper</code>	Upper limit for the posterior equal-tail interval that has the probability stipulated in the <code>prob_interval</code> argument
<code>hdi_lower</code>	Lower limit for the posterior highest-density interval that has the probability stipulated in the <code>prob_interval</code> argument
<code>hdi_upper</code>	Upper limit for the posterior highest-density interval that has the probability stipulated in the <code>prob_interval</code> argument

References

- Berger, J. O., & Wolpert, R. L. (1988). *The Likelihood Principle* (2nd ed.) Hayward, CA: Institute of Mathematical Statistics.
- Chechile, R. A. (2020). *Bayesian Statistics for Experimental Scientists: A General Introduction Using Distribution-Free Statistics*. Cambridge: MIT Press.
- Clopper, C. J., & Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26, 404-413.
- De Finetti, B. (1974). Bayesianism: Its unifying role for both the foundations and applications of statistics. *International Statistical Review/ Revue Internationale de Statistique*, 117-130.
- Ellis, R. L. (1842). On the foundations of the theory of probability. *Transactions of the Cambridge Philosophical Society*, 8, 1-6.
- Johnson, N. L., Kotz S., and Balakrishnan, N. (1995). *Continuous Univariate Distributions*, Vol. 1, New York: Wiley.
- Kolmogorov, A. N. (1933/1959). *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Berlin: Springer. English translation in 1959 as *Foundations of the Theory of Probability*. New York: Chelsea.
- Laplace, P. S. (1774). Memoire sr la probabillite des causes par les evenements. *Oeuvres complete*, 8,5-24.
- Lindley, D. V., & Phillips, L. D. (1976). Inference for a Bernoulli process (a Bayesian view). *The American Statistician*, 30, 112-119.
- Pearson, K. (1920). The fundamental problem of practical statistics. *Biometrika*, 13(1), 1-16.
- von Mises, R. (1957). *Probability, Statistics, and Truth*. New York: Dover.

See Also

[Distributions](#) for details on the functions included in the **stats** regarding the beta and the binomial distributions.

[dfba_beta_bayes_factor](#) for further documentation about the Bayes factor and its interpretation.

[dfba_beta_descriptive](#) for advanced Bayesian descriptive methods for beta distributions

Examples

```
# Example using defaults of a uniform prior and 95% interval estimates
dfba_binomial(n1 = 16,
              n2 = 2)

# Example with the Jeffreys prior and 99% interval estimates
dfba_binomial(n1 = 16,
              n2 = 2,
              a0 = .5,
              b0 = .5,
              prob_interval = .99)
```

dfba_bivariate_concordance

Bayesian Distribution-Free Correlation and Concordance

Description

Given bivariate data, computes the sample number of concordant changes nc between the two variates and the number of discordant changes nd . Provides the frequentist τ_A correlation coefficient $(nc-nd)/(nc+nd)$, and provides a Bayesian analysis of the population concordance parameter ϕ : the limit of the proportion of concordance changes between the variates. For goodness-of-fit applications, provides a concordance measure that corrects for the number of fitting parameters.

Usage

```
dfba_bivariate_concordance(
  x,
  y,
  a0 = 1,
  b0 = 1,
  prob_interval = 0.95,
  fitting.parameters = NULL
)
```

Arguments

<code>x</code>	Vector of x variable values
<code>y</code>	Vector of y variable values
<code>a0</code>	First shape parameter for the prior beta distribution (default is 1)
<code>b0</code>	Second shape parameter for the prior beta distribution (default is 1)
<code>prob_interval</code>	Desired width for interval estimates (default is .95)
<code>fitting.parameters</code>	(Optional) If either x or y values are generated by a predictive model, the number of free parameters in the model (default is NULL)

Details

The product-moment correlation depends on Gaussian assumptions about the residuals in a regression analysis. It is not robust because it is strongly influenced by any extreme outlier scores for either of the two variates. A rank-based analysis can avoid both of these limitations. The `dfba_bivariate_concordance()` function is focused on a nonparametric concordance metric for characterizing the association between the two bivariate measures.

To illustrate the nonparametric concepts of concordance and discordance, consider a specific example where there are five paired scores with

$$x = 3.8, 4.7, 4.7, 4.7, 11.8$$

and

$$y = [5.9, -4.1, 7.3, 7.3, 38.9].$$

The ranks for the x variate are 1, 3, 3, 3, 5 and the corresponding ranks for y are 2, 1, 3.5, 3.5, 5, so the five points in terms of their ranks are $P_1 = (1, 2)$, $P_2 = (3, 1)$, $P_3 = (3, 3.5)$, $P_4 = (3, 3.5)$ and

$P_5 = (5, 5)$. The relationship between any two of these points P_i and P_j , is either (1) concordant if the sign of $R_{xi} - R_{xj}$ is the same as the sign of $R_{yi} - R_{yj}$, (2) discordant if signs are different between $R_{xi} - R_{xj}$ and $R_{yi} - R_{yj}$, or (3) null if either $R_{xi} = R_{xj}$ or if $R_{yi} = R_{yj}$. For the above example, there are ten possible comparisons among the five points; six are concordant, one is discordant, and there are three comparisons lost due to ties. In general, given n bivariate scores there are $n(n-1)/2$ total possible comparisons. When there are ties in the x variate, there is a loss of T_x comparisons, when there are ties in the y variate, there are T_y lost comparisons. Ties in both x and y are denoted T_{xy} . The total number of possible comparisons, accounting for ties, is therefore:

$$n(n-1)/2 - T_x - T_y + T_{xy},$$

where T_{xy} is added to avoid double-counting of lost comparisons.

In the above example, there are three lost comparisons due to ties in x , one lost comparison due to a tie in y , and one comparison lost to a tie in both the x and y variates. Thus, there are $[(5*4)/2] - 3 - 1 + 1 = 7$ comparisons for the above example. The τ_A correlation is defined as $(n_c - n_d)/(n_c + n_d)$, which is a value on the $[-1, 1]$ interval. However, it is important to note the original developer of the frequentist τ correlation used a different coefficient that has come to be called τ_B , which is given as $(n_c - n_d)/([(n * (n-1)/2) - T_x][(n * (n-1)/2) - T_y])^{.5}$. However, τ_B does not properly correct for tied scores, which is unfortunate because τ_B is the value returned by the stats function `cor(..., method = "kendall")`. If there are no ties, then $T_x = T_y = T_{xy} = 0$ and $\tau_A = \tau_B$. But if there are ties, then the proper coefficient is given by τ_A . The `dfba_bivariate_concordance()` function provides the proper correction for tied scores and outputs a sample estimate for τ_A .

The focus for the Bayesian analysis is on the population proportion of concordance, which is the limit of the ratio $n_c/(n_c + n_d)$. This proportion is a value on the $[0, 1]$ interval, and it is called ϕ (Phi). ϕ is also connected to the population τ_A because $\tau_A = (2\phi - 1)$. Moreover, Chechile (2020) showed that the likelihood function for observing n_c concordant changes and n_d discordant changes is a censored Bernoulli process, so the likelihood is proportional to $(\phi^{n_c})(1 - \phi)^{n_d}$. In Bayesian statistics, the likelihood function is only specified as a proportional function because, unlike in frequentist statistics, the likelihood of unobserved and more extreme events are not computed. This idea is the *likelihood principle*, and its violation can lead to paradoxes (Lindley & Phillips, 1976). Also, the likelihood need only be a proportional function because the proportionality constant appears in both the numerator and denominator of Bayes theorem, so it cancels out. If the prior for ϕ is a beta distribution, then it follows that the posterior is also a beta distribution (*i.e.*, the beta is a natural Bayesian conjugate function for Bernoulli processes). The default prior for the `dfba_bivariate_concordance()` function is the flat prior (*i.e.*, $a0 = 1$ and $b0 = 1$).

In the special case where the user has a model for predicting a variate in terms of known quantities and where there are free-fitting parameters, the `dfba_bivariate_concordance()` function's concordance parameter is a goodness-of-fit measure for the scientific model. Thus, the bivariate pair are the observed value of a variate along with the corresponding predicted score from the model. The concordance proportion must be adjusted in these goodness-of-fit applications to take into account the number of free parameters that were used in the prediction model. Chechile and Barch (2021) argued that the fitting parameters increases the number of concordant changes. Consequently, the value for n_c is downward-adjusted as a function of the number of free parameters. The Chechile-Barch adjusted n_c value for a case where there are m free fitting parameters is $n_c - (n * m) + [m * (m + 1)/2]$. As an example, suppose that there are $n = 20$ scores, and the prediction equation has $m = 2$ free parameters that result in creating a prediction for each observed score (*i.e.*, there are 20 paired values of observed score x and predicted score y), and further suppose that this model results in $n_c = 170$ and $n_d = 20$. The value of n_d is kept at 20, but the number of concordant changes is reduced to $170 - (20 * 2) + (2 * 3/2) = 133$.

Value

A list containing the following components:

tau	Nonparametric Tau-A correlation
sample_p	Sample concordance proportion
nc	Number of concordant comparisons
nd	Number of discordant comparisons
a_post	The first shape parameter for the posterior beta distribution for the concordance proportion
b_post	The second shape parameter for the posterior beta distribution for the concordance proportion
a0	The first shape parameter for the prior beta distribution for the concordance proportion
b0	The second shape parameter for the prior beta distribution for the concordance proportion
prob_interval	The probability within the interval estimates for the phi parameter
post_median	Median of posterior distribution on phi
eti_lower	Lower limit of the equal-tail interval with width specified by prob_interval
eti_upper	Upper limit of the equal-tail interval with width specified by prob_interval
tau_star	Corrected tau_A to account for the number of free fitting parameter in goodness-of-fit applications
nc_star	The corrected number of concordant comparisons for a goodness-of-fit application when there is an integer value for fitting.parameters
nd_star	The number of discordant comparison when there is an integer value for fitting.parameters
sample_p_star	Correct proportion of concordant comparisons to account for free-fitting parameter for goodness-of-fit applications
a_post_star	Corrected value for the first shape parameter for the posterior for the concordance proportion when there are free fitting parameter for goodness-of-fit applications
b_post_star	The second shape parameter for the posterior distribution for the concordance proportion when there is a goodness-of-fit application
post_median_star	The posterior median for the concordance proportion when there is a goodness-of-fit application
eti_lower_star	Lower limit for the interval estimate when there is a goodness-of-fit application
eti_upper_star	Upper limit for the interval estimate when there is a goodness-of-fit application

References

- Chechile, R.A. (2020). Bayesian Statistics for Experimental Scientists: A General Introduction Using Distribution-Free Statistics. Cambridge: MIT Press.
- Chechile, R.A., & Barch, D.H. (2021). A distribution-free, Bayesian goodness-of-fit method for assessing similar scientific prediction equations. *Journal of Mathematical Psychology*. <https://doi.org/10.1016/j.jmp.2021.100800>
- Lindley, D. V., & Phillips, L. D. (1976). Inference for a Bernoulli process (a Bayesian view). *The American Statistician*, 30, 112-119.

Examples

```
x <- c(47, 39, 47, 42, 44, 46, 39, 37, 29, 42, 54, 33, 44, 31, 28, 49, 32, 37, 46, 55, 31)
y <- c(36, 40, 49, 45, 30, 38, 39, 44, 27, 48, 49, 51, 27, 36, 30, 44, 42, 41, 35, 49, 33)

dfba_bivariate_concordance(x, y)

## A goodness-of-fit example for a hypothetical case of fitting data in a
## yobs vector with prediction model

p = seq(.05,.95,.05)
ypred= 17.332 - (50.261*p) + (48.308*p^2)

# Note the coefficients in the ypred equation were found first via a
# polynomial regression

yobs<-c(19.805, 10.105, 9.396, 8.219, 6.110, 4.543, 5.864, 4.861, 6.136,
        5.789, 5.443, 5.548, 4.746, 6.484, 6.185, 6.202, 9.804, 9.332,
        14.408)

dfba_bivariate_concordance(x = yobs,
                           y = ypred,
                           fitting.parameters = 3)
```

dfba_gamma

Goodman-Kruskal Gamma

Description

Given bivariate data in the form of either a rank-ordered table or a matrix, returns the number of concordant and discordant changes between the variates, the Goodman-Kruskal gamma statistic, and a Bayesian analysis of the population concordance proportion parameter *phi*.

Usage

```
dfba_gamma(x, a0 = 1, b0 = 1, prob_interval = 0.95)
```

Arguments

x	Cross-tabulated matrix or table where cell [I, J] represents the frequency of observations where the rank of measure 1 is I and the rank of measure 2 is J.
a0	First shape parameter for the prior beta distribution (default is 1)
b0	Second shape parameter for the prior beta distribution (default is 1)
prob_interval	Desired width for interval estimates (default is 0.95)

Details

For bivariate data where two measures are restricted on an ordinal scale, such as when the two variates are ranked data over a limited set of integers, then an ordered contingency table is often a convenient data representation. For such a case the element in the $[I, J]$ cell of the matrix is the frequency of occasions where one variate has a rank value of I and the corresponding rank for the other variate is J . This situation is a special case of the more general case where there are two continuous bivariate measures. For the special case of a rank-order matrix with frequencies, there is a distribution-free concordance correlation that is in common usage: Goodman and Kruskal's gamma G (Siegel & Castellan, 1988).

Chechile (2020) showed that Goodman and Kruskal's gamma is equivalent to the more general τ_A nonparametric correlation coefficient. Historically, gamma was considered a different metric from τ because typically the version of τ in standard use was τ_B , which is a flawed metric because it does not properly correct for ties. Note: `cor(..., method = "kendall")` returns the τ_B correlation, which is incorrect when there are ties. The correct τ_A is computed by the `dfba_bivariate_concordance()` function.

The gamma statistic is equal to $(n_c - n_d)/(n_c + n_d)$, where n_c is the number of occasions when the variates change in a concordant way and n_d is the number of occasions when the variates change in a discordant fashion. The value of n_c for an order matrix is the sum of terms for each $[I, J]$ that are equal to $n_{ij}N_{ij}^+$, where n_{ij} is the frequency for cell $[I, J]$ and N_{ij}^+ is the sum of a frequencies in the matrix where the row value is greater than I and where the column value is greater than J . The value n_d is the sum of terms for each $[I, J]$ that are $n_{ij}N_{ij}^-$, where N_{ij}^- is the sum of the frequencies in the matrix where row value is greater than I and the column value is less than J . The n_c and n_d values computed in this fashion are, respectively, equal to n_c and n_d values found when the bivariate measures are entered as paired vectors into the `dfba_bivariate_concordance()` function.

As with the `dfba_bivariate_concordance()` function, the Bayesian analysis focuses on the population concordance proportion ϕ (ϕ); and $G = 2\phi - 1$. The likelihood function is proportional to $\phi^{n_c}(1 - \phi)^{n_d}$. The prior distribution is a beta function, and the posterior distribution is the conjugate beta where $a = a_0 + n_c$ and $b = b_0 + n_d$.

Value

A list containing the following components:

<code>gamma</code>	Sample Goodman-Kruskal gamma statistic; equivalent to the sample rank correlation coefficient <code>tau_A</code>
<code>a0</code>	First shape parameter for prior beta
<code>b0</code>	Second shape parameter for prior beta
<code>sample_p</code>	Sample estimate for proportion concordance $n_c/(n_c+n_d)$
<code>nc</code>	Number of concordant comparisons between the paired measures
<code>nd</code>	Number of discordant comparisons between the paired measures
<code>a_post</code>	First shape parameter for the posterior beta distribution for the ϕ parameter
<code>b_post</code>	Second shape parameter for the posterior beta distribution for the ϕ parameter
<code>post_median</code>	Median of the posterior distribution for the ϕ concordance parameter
<code>prob_interval</code>	The probability of the interval estimate for the ϕ parameter
<code>eti_lower</code>	Lower limit of the posterior equal-tail interval for the ϕ parameter where the width of the interval is specified by the <code>prob_interval</code> input
<code>eti_upper</code>	Upper limit of the posterior equal-tail interval for the ϕ parameter where the width of the interval is specified by the <code>prob_interval</code> input

References

Chechile, R.A. (2020). Bayesian Statistics for Experimental Scientists: A General Introduction Using Distribution-Free Methods. Cambridge: MIT Press.

Siegel, S., & Castellan, N. J. (1988) Nonparametric Statistics for the Behavioral Sciences. New York: McGraw Hill.

See Also

[dfba_bivariate_concordance](#) for a more extensive discussion about the τ_A statistic and the flawed τ_B correlation

Examples

```
# Example with matrix input
N <- matrix(c(38, 4, 5, 0, 6, 40, 1, 2, 4, 8, 20, 30),
            ncol = 4,
            byrow = TRUE)
colnames(N) <- c('C1', 'C2', 'C3', 'C4')
rownames(N) <- c('R1', 'R2', 'R3')
dfba_gamma(N)

# Sample problem with table input
NTable <- as.table(N)
dfba_gamma(NTable)
```

dfba_gamma_out-class *Classes for DFBA*

Description

Classes for DFBA

dfba_mann_whitney *Independent Samples Test (Mann Whitney U)*

Description

Given two independent vectors E and C, the function computes the sample Mann-Whitney U statistics U_E and U_C and provides a Bayesian analysis for the population parameter ω_E , which is the population ratio of $U_E/(U_E + U_C)$.

Usage

```
dfba_mann_whitney(
  E,
  C,
  a0 = 1,
  b0 = 1,
  prob_interval = 0.95,
  samples = 30000,
```



```

    method = NULL,
    hide_progress = FALSE
)

```

Arguments

<code>E</code>	Data for independent sample 1 ("Experimental")
<code>C</code>	Data for independent sample 2 ("Control")
<code>a0</code>	The first shape parameter for the prior beta distribution for ω_E (default is 1). Must be positive and finite.
<code>b0</code>	The second shape parameter for the prior beta distribution for ω_E (default is 1). Must be positive and finite.
<code>prob_interval</code>	Desired probability value for the interval estimate for ω_E (default is 95%)
<code>samples</code>	The number of Monte Carlo samples for ω_E when <code>method = "small"</code> (default is 30000)
<code>method</code>	(Optional) The method option is either "small" or "large". The "small" algorithm is based on a discrete Monte Carlo solution for cases where n is typically less than 20. The "large" algorithm is based on beta approximation model for the posterior distribution for the ω_E parameter. This approximation is reasonable when $n > 19$. Regardless of n , the user can stipulate method. When the method argument is omitted, the program selects the appropriate procedure
<code>hide_progress</code>	(Optional) If TRUE, hide percent progress while Monte Carlo sampling is running when <code>method = SMALL</code> . (default is FALSE).

Details

The Mann-Whitney U test is the frequentist nonparametric counterpart to the independent-groups t -test. The sample U_E statistic is the number of times that the E variate is larger than the C variate, whereas U_C is the converse number.

This test uses only rank information, so it is robust with respect to outliers, and it does not depend on the assumption of a normal model for the variates. The Bayesian version for the Mann-Whitney is focused on the population parameter ω_E , which is the population ratio $U_E / (U_E + U_C)$.

While the frequentist test effectively assumes the sharp null hypothesis that ω_E is .5, the Bayesian analysis has a prior and posterior distribution for ω_E on the $[0, 1]$ interval. The prior is a beta distribution with shape parameters a_0 and b_0 . The default is the flat prior ($a_0 = b_0 = 1$), but this prior can be altered by the user.

The `prob_interval` input is the value for probability interval estimates for ω_E . There are two cases depending on the sample size for the E and C variates. When the samples sizes are small, there is a discrete approximation method used. In this case, the Bayesian analysis considers 200 discrete values for ω_E from .0025 to .9975 in steps of .005. For each discrete value, a prior and a posterior probability are obtained. The posterior probabilities are based on Monte Carlo sampling to approximate the likelihood of obtaining the observed U_E and U_C values for each candidate value for ω_E . For each candidate value for ω_E , the likelihood for the observed sample U statistics does not depend on the true distributions of the E and C variates in the population. For each candidate ω_E , the software constructs two exponential variates that have the same ω_E value. The argument `samples` specifies the number of Monte Carlo samples used for each candidate value of ω_E .

For large sample sizes of the E and C variates, the Bayesian posterior distribution is closely approximated by a beta distribution where the shape parameters are a function of the sample U_E and

U_C statistics. The large-sample beta approximation was developed from extensive previous empirical studies designed to approximate the quantiles of the discrete approach with the corresponding quantiles for a particular beta distribution. The large- n solution also uses Lagrange polynomials for interpolation. The large- n approximation is reasonably accurate when $n > 19$ for each condition. When the method input is omitted, the function selects the appropriate procedure (*i.e.*, either the discrete case for a small sample size or the large- n approach). Nonetheless, the user can stipulate which method they desire regardless of sample size by inputting either `method="small"` or `method="large"`. The large- n solution is rapid compared to the small-sample solution, so care should be executed when choosing the `method="small"`, even for large sample sizes.

Technical details of the analysis are explained in the Chechile (2020) Communications in Statistics paper cited below.

Value

A list containing the following components:

Emean	Mean of the independent sample 1 ("Experimental") data
Cmean	Mean of the independent sample 1 ("Control") data
n_E	Number of observations of the independent sample 1 ("Experimental") data
n_C	Mean of observations of the independent sample 2 ("Control") data
U_E	Total number of comparisons for which observations from independent sample 1 ("Experimental") data exceed observations from independent sample 2 ("Control") data)
U_C	Total number of comparisons for which observations from independent sample 2 ("Control") data exceed observations from independent sample 1 ("Experimental") data)
prob_interval	User-defined width of omega_E interval estimate (default is 0.95)
a0	First shape parameter for the prior beta distribution
b0	Second shape parameter for the prior beta distribution
a_post	First shape parameter for the posterior beta distribution
b_post	Second shape parameter for the posterior beta distribution
samples	The number of desired Monte Carlo samples (default is 30000)
method	A character string indicating the calculation method used
omega_E	A vector of values representing candidate values for omega_E when method = "small"
omegapost	A vector of values representing discrete probabilities for candidate values of omega_E
priorvector	A vector of values representing prior discrete probabilities of candidate values of omega_E when method = "small"
priorprH1	Prior probability of the alternative model that omega_E exceeds 0.5
prH1	Posterior probability of the alternative model that omega_E exceeds 0.5
BF10	Bayes Factor describing the relative increase in the posterior odds for the alternative model that omega_E exceeds 0.5 over the null model of omega_E less than or equal to 0.5
omegabar	Posterior mean estimate for omega_E
eti_lower	Lower limit of the equal-tail probability interval for omega_E with probability width indicated by prob_interval

eti_upper	Upper limit of the equal-tail probability interval for omega_E with probability width indicated by prob_interval
hdi_lower	Lower limit of the highest-density probability interval for omega_E with probability width indicated by prob_interval when method = "small"
hdi_upper	Upper limit of the highest-density probability interval for omega_E with probability width indicated by prob_interval when method = "small"

References

Cechile, R.A. (2020). Bayesian Statistics for Experimental Scientists: A General Introduction Using Distribution-Free Methods. Cambridge: MIT Press.

Cechile, R.A. (2020). A Bayesian analysis for the Mann-Whitney statistic. Communications in Statistics – Theory and Methods 49(3): 670-696. <https://doi.org/10.1080/03610926.2018.1549247>.

Examples

```
# Note: examples with method = "small" have long runtimes due to Monte Carlo
# sampling; please feel free to run them in the console.

# Examples with large n per group
# The data for each condition are presorted only for the user convenience if
# checking the U stats by hand

groupA <- c(43, 45, 47, 50, 54, 58, 60, 63, 69, 84, 85, 91, 99, 127, 130,
            147, 165, 175, 193, 228, 252, 276)
groupB <- c(0, 01, 02, 03, 05, 14, 15, 23, 23, 25, 27, 32, 57, 105, 115, 158,
            161, 181, 203, 290)

dfba_mann_whitney(E = groupA,
                  C = groupB)

# The following uses a Jeffreys prior instead of a default flat prior:
dfba_mann_whitney(E = groupA,
                  C = groupB,
                  a0 = .5,
                  b0 = .5)

# The following also uses a Jeffreys prior but the analysis reverses the
# variates:
dfba_mann_whitney(E = groupB,
                  C = groupA,
                  a0 = .5,
                  b0 = .5)

# Note that BF10 from the above analysis is 1/BF10 from the original order
# of the variates.

# The next analysis constructs 99% interval estimates with the Jeffreys
# prior.

dfba_mann_whitney(E = groupA,
                  C = groupB,
                  a0 = .5,
                  b0 = .5,
```

```

        prob_interval=.99)

# The following forces a discrete approach with a flat prior for a case with
# large n:

## Not run:
dfba_mann_whitney(E = groupA,
                  C = groupB,
                  method = "small",
                  hide_progress = FALSE)

## End(Not run)

#Examples with small n per group

groupC <- c(96.49, 96.78, 97.26, 98.85, 99.75, 100.14, 101.15, 101.39,
            102.58, 107.22, 107.70, 113.26)
groupD <- c(101.16, 102.09, 103.14, 104.70, 105.27, 108.22, 108.32, 108.51,
            109.88, 110.32, 110.55, 113.42)

## Not run:
CDex1<-dfba_mann_whitney(E = groupC,
                        C = groupD,
                        hide_progress = FALSE)

CDex1

CDex2<-dfba_mann_whitney(E = groupC,
                        C = groupD,
                        samples = 50000,
                        hide_progress = FALSE)

CDex2

CDex3<-dfba_mann_whitney(E = groupC,
                        C = groupD,
                        hide_progress = FALSE)

CDex3

## End(Not run)
# Note that CDex1 and CDex2 are replication analyses for the discrete approach.
# The variability is due to the different outcomes from the Monte Carlo
# sampling.

# Plot output
# with prior and posterior curves

## Not run:

plot(CDex1)

# with only posterior curve
plot(CDex2,
     plot.prior = FALSE)

```

```
## End(Not run)
```

dfba_mcnemar

Bayesian Repeated-Measures McNemar Test for Change

Description

Given a randomized-block or repeated-measures design where the response is coded as either 0 or 1, examines the subset of cases where there is a change in the response between the two measurements and provides a Bayesian analysis of the population change rate ϕ_{rb} (ϕ_{rb}) between the two measurements.

Usage

```
dfba_mcnemar(n_01, n_10, a0 = 1, b0 = 1, prob_interval = 0.95)
```

Arguments

n_01	The number of cases where the first response is 0 and the second response is 1.
n_10	The number of cases where the first response is 1 and the second response is 0.
a0	The first shape parameter for the prior beta distribution for the ϕ_{rb} parameter. Must be positive and finite.
b0	The second shape parameter for the prior beta distribution for the ϕ_{rb} parameter. Must be positive and finite.
prob_interval	Desired probability for interval estimates for ϕ_{rb} (default is .95).

Details

Sometimes, researchers are interested in the detection of a change in the response rate pre- and post-treatment. The frequentist McNemar test is a nonparametric test that examines the subset of binary categorical responses where the response changes between the two tests (Siegel & Castellan, 1988). The frequentist test assumes the null hypothesis that the change rate is 0.5. Chechile (2020) pointed out that the subset of change cases are binomial data, so a Bayesian analysis can be done for the population response-switching rate ϕ_{rb} (styled ϕ_{rb} elsewhere in the documentation for this function). Both the prior and posterior distribution for ϕ_{rb} are beta distributions.

The user should be aware that the McNemar test is a change-detection assessment of a binary response. To illustrate this fact, consider the hypothetical case of a sample of 50 people who evaluate two political candidates before and after a debate. Suppose 26 people prefer Candidate A both before and after the debate and 14 people prefer Candidate B both before and after the debate, but 9 people switch their preference from Candidate A to Candidate B and 1 person switches their preference from Candidate B to Candidate A. Despite the fact that this sample has 50 participants, it is only the 10 people who switch their preference that are being analyzed with the McNemar test. Among this subset, there is evidence that Candidate B did better on the debate. Overall, support for Candidate A in the whole sample fell from 35 out of 50 (70%) to 27 out of 50 (54%): still a majority, but a smaller one than Candidate A enjoyed prior to the debate.

The `dfba_mcnemar()` function requires two inputs, `n_01` and `n_10`, which are, respectively, the number of $0 \rightarrow 1$ changes and the number of $1 \rightarrow 0$ switches in the binary responses between the two tests. Since the cases where there is a switch are binomial trials, the prior and posterior

distributions for ϕ_{rb} are beta distributions. The prior distribution shape parameters are a_0 and b_0 . The default prior is a uniform distribution (*i.e.*, $a_0 = b_0 = 1$). The `prob_interval` argument stipulates the probability within the equal-tail interval limits for ϕ_{rb} . The default value for that argument is `prob_interval = .95`.

Besides computing the posterior mean, posterior median, equal-tail interval limits, and the posterior probability that $\phi_{rb} > .5$, the function also computes two Bayes factor values. One is the *point* Bayes factor BF10 against the null hypothesis that $\phi_{rb} = 0.5$. The second Bayes factor BF10 is the *interval* Bayes factor against the null hypothesis that $\phi_{rb} \leq 0.5$. If the interval Bayes factor BF10 is very low, then there is support to some degree for the null hypothesis that $\phi_{rb} < 0.5$. In this case the Bayes factor BF01 in support of the interval null hypothesis is given by $BF01 = 1/BF10$.

Value

A list containing the following components:

<code>n_01</code>	The number of cases where the first response is 0 and the second response is 1
<code>n_10</code>	The number of cases where the first response is 1 and the second response is 0
<code>prob_interval</code>	Desired posterior probability within the equal-tail interval limits for <code>phi_rb</code>
<code>a0</code>	The first shape parameter for the prior beta distribution for the <code>phi_rb</code> parameter
<code>b0</code>	The second shape parameter for the prior beta distribution for the <code>phi_rb</code> parameter
<code>a_post</code>	First shape parameter for the posterior beta distribution for the <code>phi_rb</code> parameter
<code>b_post</code>	Second shape parameter for the posterior beta distribution for the <code>phi_rb</code> parameter
<code>post_mean</code>	Posterior mean for <code>phi_rb</code>
<code>post_median</code>	Posterior median for <code>phi_rb</code>
<code>eti_lower</code>	Lower limit for the posterior equal-tail interval estimate for <code>phi_rb</code> that contains the probability defined in <code>prob_interval</code>
<code>eti_upper</code>	Upper limit for the posterior equal-tail interval estimate for <code>phi_rb</code> that contains the probability defined in <code>prob_interval</code>
<code>BF10point</code>	The Bayes factor against the point null hypothesis that <code>phi_rb = .5</code>
<code>BF10interval</code>	The Bayes factor against the interval null hypothesis that <code>phi_rb</code> is less than or equal to <code>.5</code>
<code>postH1</code>	The posterior probability that <code>phi_rb > .5</code>

References

- Chechile, R.A. (2020). Bayesian Statistics for Experimental Scientists: A General Introduction Using Distribution-Free Methods. Cambridge: MIT Press.
- Siegel, S., & Castellan, N. J. (1988) Nonparametric Statistics for the Behavioral Sciences. New York: McGraw Hill.

See Also

[dfba_beta_bayes_factor](#) for further documentation about the Bayes factor and its interpretation.

Examples

```
## Examples with default value for a0, b0 and prob_interval

dfba_mcnemar(n_01 = 17,
             n_10 = 2)

## Using the Jeffreys prior and .99 equal-tail interval

dfba_mcnemar(n_01 = 17,
             n_10 = 2,
             a0 = .5,
             b0 = .5,
             prob_interval = .99)
```

dfba_median_test	<i>Bayesian Median Test</i>
------------------	-----------------------------

Description

Given two independent groups of continuous variables, performs a Bayesian analysis of the likelihood of observing an above-median value from one of the groups relative to expectation.

Usage

```
dfba_median_test(E, C, a0 = 1, b0 = 1)
```

Arguments

E	Numeric vector of values for the continuous measurements for group 1 (generically denoted E for <i>Experimental</i> group).
C	Numeric vector of values for the continuous measurements for group 2 (generically denoted C for <i>Control</i> group).
a0	The first shape parameter for the prior beta distribution for the binomial parameter ϕ (default is 1). Must be positive and finite.
b0	The second shape parameter for the prior beta distribution for the binomial parameter ϕ (default is 1). Must be positive and finite.

Details

Given continuous measurements E and C from two separate and independent groups, a combined sample median value can be computed. For the frequentist median test, a 2x2 table is created. Row 1 consists of the frequencies of the above-median responses in terms of the two groups (*i.e.*, n_{Eabove} and n_{Cabove}). Row 2 has the respective frequencies for the values that are at or below the combined median (*i.e.*, n_{Ebelow} and n_{Cbelow}). See Siegel & Castellan (1988) for the details concerning the frequentist median test.

Chechile (2020) provided an alternative Bayesian analysis for the median-test procedure of examining continuous data in terms of the categorization of the values as being either above the combined median or not. The frequencies in row 1 (above median response) are binomial frequencies in terms of the group origin (*i.e.*, E versus C). From a Bayesian perspective, a population-level ϕ parameter

can be defined for the population proportion of E values that are above the combined sample median. Similarly, the frequencies for the scores at or below the combined sample median can also be examined; in that case, the corresponding population proportion in the E condition must be $1 - \phi$. Thus, it is sufficient only to examine the above-median frequencies to make an inference about the ϕ parameter. Since this is a binomial problem, the prior and posterior distributions for the population ϕ parameter belong to the beta family of distributions. The default prior for this function is the uniform distribution, *i.e.*, $a_0 = b_0 = 1$. The posterior shape parameters for ϕ are $a_{\text{post}} = a_0 + nE_{\text{above}}$ and $b_{\text{post}} = b_0 + nC_{\text{above}}$.

Because the number of scores in groups E and C might be very different, it is important to examine the ϕ parameter relative to an expected base-rate value from the sample. For example, suppose that there are $nE = 90$ values from the E group and $nC = 10$ values from the C group. In this example, there are 50 scores that are above the combined median (and no ties that would result in fewer than half of the scores being greater than the median) that should be examined to see if ϕ is greater than 0.9. If there were no difference between the E and C conditions whatsoever in this hypothetical example, then about 90 percent of the above-median values would be from the E group. If the posterior ϕ parameter were substantially above the group E base rate, then that would support the hypothesis that group E has larger values than group C in the population.

The `dfba_median_test()` provides the descriptive sample information for the combined median as well as the entries for a table for the frequencies for the E and C scores that are above the median, as well as the frequencies for the E and C scores at or below the median. The function also provides the prior and posterior probabilities that the E and C groups exceeding their respective base rates for a value being above the median. The function also evaluates the hypotheses that the E and C response rates for the above-median responses exceeding their base rate. Bayes factors are provided for these hypothesis.

Because the Bayesian median test ignores the available rank-order information, this procedure has less power than the Bayesian Mann-Whitney analysis that can be computed for the same data. Nonetheless, sometimes researchers are interested if condition differences are so strong that even a lower power median test can detect the difference.

Value

A list containing the following components:

<code>median</code>	The sample combined median for the E and C values
<code>nE</code>	The number of scores from group E
<code>nC</code>	The number of scores from group C
<code>Ebaserate</code>	The proportion $nE/(nE+nC)$
<code>Cbaserate</code>	The proportion $nC/(nE+nC)$
<code>nEabove</code>	Number of E responses above the median
<code>nCabove</code>	Number of C responses above the median
<code>nEbelow</code>	Number of E responses at or below median
<code>nCbelow</code>	Number of C response at or below median
<code>a0</code>	The first shape parameter for the prior beta distribution for the population binomial parameter
<code>b0</code>	The second shape parameter for the prior beta distribution for the population binomial parameter
<code>a_post</code>	Posterior first shape parameter for the beta distribution for the probability that an above-median response is from the E group

b_post	Posterior second shape parameter for the beta distribution for the probability that an above-median response is from the E group
postEhi	Posterior probability that an above-median response exceeds the E group base rate
postChi	Posterior probability that an above-median response exceeds the C group base rate
priorEhi	The probability that a beta prior distribution would exceed the E group base rate
priorChi	The probability that a beta prior distribution would exceed the C group base rate
BF10E	The Bayes factor in favor of the hypothesis that an above-median response from the E group is more probable than the E expected base rate
BF10C	The Bayes factor in favor of the hypothesis that an above-median response from the C group is more probable than the C group base rate

References

- Chechile, R.A. (2020). Bayesian Statistics for Experimental Scientists: A General Introduction Using Distribution-Free Methods. Cambridge: MIT Press.
- Siegel, S., & Castellan, N. J. (1988) Nonparametric Statistics for the Behavioral Sciences. New York: McGraw Hill.

See Also

[dfba_beta_bayes_factor](#) for further documentation about the Bayes factor and its interpretation.

[dfba_mann_whitney](#) for a more powerful alternative Bayesian analysis of the *E* and *C* values that use rank order information.

Examples

```
## Example with the default uniform prior
group1 <- c(12.90, 10.84, 22.67, 10.64, 10.67, 10.79, 13.55, 10.95, 12.19,
            12.76, 10.89, 11.02, 14.27, 13.98, 11.52, 13.49, 11.22, 15.07,
            15.74, 19.00)

group2 <- c(4.63, 58.64, 5.07, 4.66, 4.13, 3.92, 3.39, 3.57, 3.56, 3.39)

dfba_median_test(E = group1,
                 C = group2)

## Example with the Jeffreys prior
dfba_median_test(group1,
                 group2,
                 a0 = .5,
                 b0 = .5)
```

dfba_power_curve *Power Curves*

Description

This function is a design tool for comparing Bayesian distribution-free power with frequentist t power for a range of delta values, which are the separation values between two continuous variates that can be randomly sampled from one of nine different probability models. The function provides a table of power values for a fixed sample size value of n .

Usage

```
dfba_power_curve(
  n = 20,
  a0 = 1,
  b0 = 1,
  delta_step = 0.05,
  model,
  design,
  effect_crit = 0.95,
  shape1 = 1,
  shape2 = 1,
  block_max = 0,
  samples = 1000,
  hide_progress = FALSE
)
```

Arguments

n	The sample size for both variates (default is 20)
a0	The first shape parameter for the prior beta distribution (default is 1). Must be positive and finite.
b0	The second shape parameter for the prior beta distribution (default is 1). Must be positive and finite.
delta_step	The increment between successive delta values, which range from 0 to 20*delta_step (default value is .05)
model	Theoretical probability model for the data. One of "normal", "weibull", "cauchy", "lognormal", "chisquare", "logistic", "exponential", "gumbel", or "pareto".
design	Indicates the data structure. One of "independent" or "paired".
effect_crit	Stipulated value for a significant differences for a t -test ($1 - p$), and the critical probability for the Bayesian alternative hypothesis for a Bayesian distribution-free analysis
shape1	The shape parameter for the condition 1 variate for the distribution indicated by the model input (default is 1)
shape2	The shape parameter for the condition 2 variate for the distribution indicated by the model input (default is 1)
block_max	The maximum size for a block effect (default is 0)

<code>samples</code>	Desired number of Monte Carlo data sets drawn to estimate the power (default is 1000)
<code>hide_progress</code>	(Optional) If TRUE, hide percent progress while Monte Carlo sampling is running. (default is FALSE).

Details

Researchers need to make experimental-design decisions such as the choice about the sample size per condition and the decision of whether to use a within-block design or an independent-group design. These planning issues arise regardless if one uses either a frequentist or Bayesian approach to statistical inference. In the DFBA package there are a number of functions to help users with these decisions. The `dfba_power_curve()` function produces the Bayesian power estimate from a distribution-free analysis along with the corresponding frequentist power from a parametric *t*-test, for 21 delta values, which range from 0 to $20 \times \text{delta_step}$ where delta is the separation between two random variates. The sample size for the power estimates is the same value *n* in each condition. The power estimates are based on a number of Monte Carlo sampled data sets generated by the `dfba_sim_data()` function.

For each data set, statistical tests are performed. If `design = "paired"`, the frequentist *t*-test is a one-tailed test on the within-block difference scores to assess the null hypothesis that the population mean for E is greater than the population mean for C; if `design = "independent"`, the frequentist *t*-test is the one-tailed test to assess if there is a significant difference between the two independent conditions (*i.e.* if the mean for condition 2 is significantly greater than the condition 1 mean). If `design = "paired"`, the Bayesian analysis assesses if the posterior probability for $\phi_w > .5$ on the Bayesian Wilcoxon test is greater than `effect_crit`; if `design = "independent"`, the Bayesian analysis assesses if the posterior probability for $\omega_E > .5$ on a Bayesian Mann-Whitney test is greater than `effect_crit`. The frequentist power is estimated by the proportion of the data sets where a parametric *t*-test detects a significant effect because the upper-tail *t* value has a *p*-value less than $1 - \text{effect_crit}$. The Bayesian power is the proportion of the data sets where a posterior probability for the alternative hypothesis is greater than `effect_crit`. The default value for the `effect_crit` argument is .95. The frequentist *p*-value and the Bayesian posterior probability for the alternative hypothesis are calculated using the `dfba_sim_data()` function.

The arguments for the `dfba_power_curve()` function are passed into the `dfba_sim_data()` function. Besides the sample size *n*, there are eight other arguments that are required by the `dfba_power_curve()` function, which are passed into the `dfba_sim_data()` function:

- `a0`
- `b0`
- `model`
- `design`
- `delta`
- `shape1`
- `shape2`
- `block_max`.

The `a0` and `b0` values are the respective first and second beta shape parameters for the prior distribution needed for the Bayesian distribution-free tests, which are ultimately done by calling either the `dfba_wilcoxon()` function or by the `dfba_mann_whitney()`.

The `model` argument is one of the following strings:

- `"normal"`

nsims	The number of Monte Carlo data sets; equal to the value of the <code>samples</code> argument
model	Probability model for the data
design	The design for the data; one of "independent" or "paired"
a0	The first shape parameter for the beta prior distribution for the Bayesian analysis
b0	The second shape parameter for the beta prior distribution for the Bayesian analysis
effect_crit	The criterion probability for considering a posterior probability for the alternative hypothesis to be a detection; it is also $1 - p_crit$ for a frequentist t -test
block_max	The maximum size of a block effect; equal to the <code>block_max</code> argument
delta_vec	Vector of the 21 delta offset values
Bayes_power	The vector of 21 Bayesian power values
t_power	The vector of 21 frequentist power from t -tests
outputdf	A dataframe for the Bayesian power and the corresponding frequentist t power as a function of delta

References

- Arnold, B. C. (1983). Pareto Distribution. Fairland, MD: International Cooperative Publishing House.
- Chechile, R. A. (2017). A Bayesian analysis for the Wilcoxon signed-rank statistic. *Communications in Statistics - Theory and Methods*, <https://doi.org/10.1080/03610926.2017.1388402>
- Chechile, R. A. (2020). A Bayesian analysis for the Mann-Whitney statistic. *Communications in Statistics - Theory and Methods*, <https://doi.org/10.1080/03610926.2018.1549247>
- Fishman, G. S. (1996) Monte Carlo: Concepts, Algorithms and Applications. New York: Springer.
- Hardy, M. (2010). Pareto's Law. *Mathematical Intelligencer*, 32, 38-43.
- Johnson, N. L., Kotz S., and Balakrishnan, N. (1995). Continuous Univariate Distributions, Vol. 1, New York: Wiley.
- Pareto, V. (1897). *Cours d'Economie Politique*. Vol. 2, Lausanne: F. Rouge.

See Also

[Distributions](#) for details on the parameters of the normal, Weibull, Cauchy, lognormal, chi-squared, logistic, and exponential distributions.

[dfba_wilcoxon](#)

[dfba_mann_whitney](#)

[dfba_sim_data](#) for further details about the data for two conditions that differ in terms of their theoretical mean by an amount delta.

Examples

```
# Note: these examples have long runtimes due to Monte Carlo sampling;
# please feel free to run them in the console.
```

```
## Not run:
dfba_power_curve(n = 85,
                 model = "normal",
```

```

        design = "independent",
        hide_progress = FALSE)

dfba_power_curve(n = 85,
                 model = "normal",
                 design = "paired",
                 hide_progress = FALSE)

dfba_power_curve(n = 85,
                 model = "normal",
                 design = "paired",
                 delta_step = .03,
                 hide_progress = FALSE)

dfba_power_curve(n = 30,
                 model = "lognormal",
                 design = "independent",
                 delta_step = .06,
                 block_max = 3,
                 samples = 2500,
                 hide_progress = FALSE)

dfba_power_curve(n = 30,
                 model = "lognormal",
                 design = "paired",
                 delta_step = .06,
                 block_max = 3,
                 samples = 2500,
                 hide_progress = FALSE)

# Using the Jeffreys prior rather than default flat prior

dfba_power_curve(n = 30,
                 model = "lognormal",
                 design = "independent",
                 a0 = .5,
                 b0 = .5,
                 delta_step = .06,
                 block_max = 3,
                 samples = 2500,
                 hide_progress = FALSE)

## End(Not run)

```

dfba_sign_test

Bayesian Sign Test

Description

Given two paired continuous variates Y1 and Y2, provides a Bayesian sign test to assess the positivity rate for the difference $Y1 - Y2$.

Usage

```
dfba_sign_test(Y1, Y2, a0 = 1, b0 = 1, prob_interval = 0.95)
```

Arguments

Y1	Vector of the continuous measurements for one group
Y2	Vector of the continuous values paired with the Y1 vector for the values in a second group
a0	The first shape parameter for the prior beta distribution for the positive-sign rate parameter (default is 1). Must be positive and finite.
b0	The second shape parameter for the prior beta distribution for the positive-sign rate parameter (default is 1). Must be positive and finite.
prob_interval	Desired probability within interval limits for interval estimates of the positivity rate parameter (default is .95)

Details

Given two paired continuous variates Y_1 and Y_2 for two repeated measures, statistical tests for differences examine the difference measure $d = Y_1 - Y_2$. The t -test is a conventional frequentist parametric procedure to assess values of d . There are also two common frequentist nonparametric tests for assessing condition differences: the sign test and the Wilcoxon signed-rank test. The sign test is less powerful than the Wilcoxon signed-rank test (Siegel & Castellan, 1988). The appeal of the sign test, for some researchers, is that it is simple and - in some cases - sufficient for demonstrating strong differences.

The `dfba_sign_test()` function provides a Bayesian version of the sign test (the function `dfba_wilcoxon()` provides the Bayesian signed-rank test). While the Wilcoxon procedure uses both rank and sign information, the sign test uses only sign information. The `dfba_sign_test()` function finds the number of positive and negative d values, which appear in the output as `n_pos` and `n_neg`, respectively. Note that it is standard both in the frequentist sign test and in the frequentist Wilcoxon signed-rank procedure to remove the d values that are zero. Consequently, the signs for the nonzero d values are binary, so the posterior is a beta distribution with shape parameters a - denoted in the output as `a_post` and b - denoted in the output as `b_post` - where `a_post` = `a0` + `n_pos` and `b_post` = `b0` + `n_neg` and `a0` and `b0` are the respective first and second beta shape parameters for the prior distribution. The default prior is a uniform distribution `a0` = `b0` = 1.

The function estimates the population rate for positive signs by calling `dfba_beta_descriptive()` using the computed `a_post` and `b_post` as arguments. Since interest in the sign test is focused on the null hypothesis that the positivity rate is less than or equal to .5, `dfba_sign_test()` calls `dfba_beta_bayes_factor()` to calculate the prior and posterior probabilities for the alternative hypothesis that the positivity rate is greater than .5. The output also includes the Bayes factors `BF10` and `BF01`, where `BF01` = `1/BF10`. Large values of `BF01` indicate support for the null hypothesis; large values of `BF10` indicate support for the alternative hypothesis.

Value

A list containing the following components:

Y1	Vector of continuous values for the first within-block group
Y2	Vector of continuous values for the second within-block group
a0	First shape parameter for the prior beta distribution for the population parameter for the positivity rate
b0	Second shape parameter for the prior beta distribution for the population positivity rate
prob_interval	The probability within the interval limits for the interval estimate of population positivity rate

n_pos	Sample number of positive differences
n_neg	Sample number of negative differences
a_post	First shape parameter for the posterior beta distribution for the population positivity rate
b_post	Second shape parameter for the posterior beta distribution for the population positivity rate for differences
phimean	Mean of the posterior distribution for the positivity rate parameter
phimedian	Median of the posterior distribution for the positivity rate parameter
phimode	Mode of the posterior distribution for the positivity rate parameter
eti_lower	Lower limit of the equal-tail interval estimate of the positivity rate parameter
eti_upper	Upper limit of the equal-tail interval estimate of the positivity rate parameter
hdi_lower	Lower limit for the highest-density interval estimate of the positivity rate parameter
hdi_upper	Upper limit for the highest-density interval estimate of the positivity rate parameter
post_H1	Posterior probability that the positivity rate is greater than .5
prior_H1	Prior probability that the positivity rate is greater than .5
BF10	Bayes factor in favor of the alternative hypothesis that the positivity rate is greater than .5
BF01	Bayes factor in favor of the null hypothesis that the positivity rate is equal to or less than .5

References

- Chechile, R. A. (2020). Bayesian Statistics for Experimental Scientists: A General Introduction Using Distribution-Free Methods. Cambridge, MIT Press.
- Siegel, S., & Castellan, N. J. (1988). Nonparametric Statistics for the Behavioral Sciences. New York: McGraw Hill.

See Also

[dfba_beta_descriptive](#) for details on the descriptive statistics in the output

[dfba_beta_bayes_factor](#) for details on Bayes Factors calculated on the basis of beta distributions

[dfba_wilcoxon](#) for an alternative, more powerful Bayesian nonparametric test for evaluating repeated-measures data.

Examples

```
measure_1 <- c(1.49, 0.64, 0.96, 2.34, 0.78, 1.29, 0.72, 1.52,
              0.62, 1.67, 1.19, 0.860)

measure_2 <- c(0.53, 0.55, 0.58, 0.97, 0.60, 0.22, 0.05, 13.14,
              0.63, 0.33, 0.91, 0.37)

dfba_sign_test(Y1 = measure_1,
               Y2 = measure_2)

dfba_sign_test(measure_1,
```



```

measure_2,
a0 = .5,
b0 = .5,
prob_interval = .99)

```

dfba_sim_data

Simulated Data Generator and Inferential Comparison

Description

This function is designed to be called by other DFBA programs that compare frequentist and Bayesian power. The function generates simulated data for two conditions that can be from nine different probability models. The program also computes the frequentist p -value from a t -test on the generated data, and it computes the Bayesian posterior probability from a distribution-free analysis of the difference between the two conditions.

Usage

```

dfba_sim_data(
  n = 20,
  a0 = 1,
  b0 = 1,
  model,
  design,
  delta,
  shape1 = 1,
  shape2 = 1,
  block_max = 0
)

```

Arguments

n	Number of values per condition
a0	The first shape parameter for the prior beta distribution (default is 1). Must be positive and finite.
b0	The second shape parameter for the prior beta distribution (default is 1). Must be positive and finite.
model	Theoretical probability model for the data. One of "normal", "weibull", "cauchy", "lognormal", "chisquare", "logistic", "exponential", "gumbel", or "pareto"
design	Indicates the data structure. One of "independent" or "paired".
delta	Theoretical mean difference between conditions; the second condition minus the first condition
shape1	The shape parameter for condition 1 for the distribution indicated by model input (default is 1)
shape2	The shape parameter for condition 2 for the distribution indicated by model input (default is 1)
block_max	The maximum size for a block effect (default is 0)

Details

Researchers need to make experimental-design decisions such as the choice about the sample size per condition and the decision to use a within-block design or an independent-group design. These planning issues arise regardless if one uses either a frequentist or Bayesian approach to statistical inference. In the DFBA package, there are a number of functions to help users with these decisions.

The `dfba_sim_data()` program is used along with other functions to assess the relative power for detecting a condition difference of an amount δ between two conditions. δ is an input for the `dfba_sim_data()` program, and it must be a nonnegative value. Specifically, the `dfba_sim_data()` program generates two sets of data that are randomly drawn from one of nine different theoretical models. The input 'model' stipulates the data generating probability function. The input 'model' is one of the following strings:

- "normal"
- "weibull"
- "cauchy"
- "lognormal"
- "chisquare"
- "logistic"
- "exponential"
- "gumbel"
- "pareto"

For each model there are n continuous scores randomly sampled for each condition, where n is a user-specified input value. The design argument is either "independent" or "paired", and stipulates whether the two sets of scores are either independent or from a common blocks such as for the case of two scores for the same person (*i.e.*, one in each condition).

The `shape1` and `shape2` arguments are values for the shape parameter for the respective first and second condition, and their meaning depends on the probability model. For `model="normal"`, these parameters are the standard deviations of the two distributions. For `model="weibull"`, the parameters are the Weibull shape parameters. For `model="cauchy"`, the parameters are the scale factors for the Cauchy distributions. For `model="lognormal"`, the shape parameters are the standard deviations for $\log(X)$. For `model="chisquare"`, the parameters are the degrees of freedom (df) for the two distributions. For `model="logistic"`, the parameters are the scale factors for the distributions. For `model="exponential"`, the parameters are the rate parameters for the distributions.

For the Gumbel distribution, the E variate is equal to $\delta - \text{shape2} \cdot \log(\log(1/U))$ where U is a random value sampled from the uniform distribution on the interval $[.00001, .99999]$, and the C variate is equal to $-\text{shape1} \cdot \log(\log(1/U))$ where U is another score sampled from the uniform distribution. The `shape1` and `shape2` arguments for `model="gumbel"` are the scale parameters for the distributions. The Pareto model is a distribution designed to account for income distributions as studied by economists (Pareto, 1897). For the Pareto distribution, the cumulative function is equal to $1 - (x_m/x)^\alpha$ where x is greater than x_m (Arnold, 1983). In the E condition, $x_m = 1 + \delta$ and in the C condition $x_m = 1$. The α parameter is 1.16 times the shape parameters `shape1` and `shape2`. Since the default value for each shape parameter is 1, the resulting α value of 1.16 is the default value. When $\alpha = 1.16$, the Pareto distribution approximates an income distribution that represents the 80-20 law where 20% of the population receives 80% of the income (Hardy, 2010).

The `block_max` argument provides for incorporating block effects in the random sampling. The block effect for each score is a separate effect for the block. The block effect B for a score is a

random number drawn from a uniform distribution on the interval $[0, \text{block_max}]$. When `design = "paired"`, the same random block effect is added to the score in the first condition, which is the random C value, and it is also added to the corresponding paired value for the E variate. Thus, the pairing research design eliminates the effect of block variation for the assessment of condition differences. When `design = "independent"`, there are different block-effect contributions to the E and C variates, which reduces the discrimination of condition differences because it increases the variability of the difference in the two variates. The user can study the effect of the relative discriminability of detecting an effect of delta by adjusting the value of the `block_max` argument. The default for `block_max` is 0, but it can be altered to any non-negative real number.

The output from calling the `dfba_sim_data()` function are two statistics that are based on the n scores generated in the two conditions. One statistic is the frequentist p -value for rejecting the null hypothesis that $\delta \leq 0$ from a parametric t -test. The p -value is the upper tail probability of the sample t -statistic for either the paired t -test when `design = "paired"` or it is the upper tail probability of the sample t -statistic for the two-group t -test when `design = "independent"`. The second output statistic is the Bayesian posterior probability for one of two possible nonparametric tests. If `design = "paired"`, the `dfba_sim_data()` function calls the `dfba_wilcoxon()` function to ascertain the posterior probability that $\phi_w > .5$. If `design = "independent"`, the `dfba_sim_data()` function calls the `dfba_mann_whitney()` function to estimate the posterior probability that $\omega_E > .5$. The arguments `a0` and `b0` for the `dfba_sim_data()` function are passed along to either the `dfba_wilcoxon()` function or the `dfba_mann_whitney()` function. The default values are `a0 = b0 = 1`.

Value

A list containing the following components:

<code>pvalue</code>	The upper tail of the sample t value for the test that $\delta \leq 0$
<code>prH1</code>	Bayesian posterior probability either for the hypothesis that $\phi_w > .5$ from the nonparametric Wilcoxon test when <code>design = "paired"</code> or for the hypothesis that $\omega_E > .5$ from the Mann-Whitney test when <code>design = "independent"</code>
<code>C</code>	Vector of length n of simulated values for condition 1
<code>E</code>	Vector of length n of simulated values for condition 2
<code>design</code>	The data structure indicated by the <code>design</code> argument. One of <code>"independent"</code> or <code>"paired"</code> .

Note

Random sampling for both the Gumbel and the Pareto distributions are generated by the `dfba_sim_data()` function using the inverse transform method (Fishman, 1996).

References

- Arnold, B. C. (1983). Pareto Distribution. Fairland, MD: International Cooperative Publishing House.
- Chechile, R. A. (2017). A Bayesian analysis for the Wilcoxon signed-rank statistic. Communications in Statistics - Theory and Methods, <https://doi.org/10.1080/03610926.2017.1388402>.
- Chechile, R. A. (2020). A Bayesian analysis for the Mann-Whitney statistic. Communications in Statistics - Theory and Methods, <https://doi.org/10.1080/03610926.2018.1549247>.
- Fishman, G. S. (1996) Monte Carlo: Concepts, Algorithms and Applications. New York: Springer.
- Hardy, M. (2010). Pareto's Law. Mathematical Intelligencer, 32, 38-43.

Johnson, N. L., Kotz S., and Balakrishnan, N. (1995). Continuous Univariate Distributions, Vol. 1, New York: Wiley.

Pareto, V. (1897). Cours d'Economie Politique. Vol. 2, Lausanne: F. Rouge.

See Also

[Distributions](#) for details on the parameters of the normal, Weibull, Cauchy, lognormal, chi-squared, logistic, and exponential distributions.

[dfba_wilcoxon](#)

[dfba_mann_whitney](#)

Examples

```
# Example of two paired normal distributions where the s.d. of the two
# conditions are 1 and 4.

dfba_sim_data(n = 50,
              model = "normal",
              design = "paired",
              delta = .4,
              shape1 = 1,
              shape2 = 4)

# Example of two independent Weibull variates with their shape parameters =.8
# and with a .25 offset

dfba_sim_data(n = 80,
              model = "weibull",
              design = "independent",
              delta = .25,
              shape1 = .8,
              shape2 = .8)

# Example of two independent Weibull variates with their shape
# parameters = .8 and with a .25 offset along with some block differences
# with the max block effect being 1.5

dfba_sim_data(n = 80,
              model = "weibull",
              design = "independent",
              delta = .25,
              shape1 = .8,
              shape2 = .8,
              block_max = 1.5)

# Example of two paired Cauchy variates with a .4 offset

dfba_sim_data(n = 50,
              model = "cauchy",
              design = "paired",
              delta = .4)

# Example of two paired Cauchy variates with a .4 offset where the Bayesian
# analysis uses the Jeffreys prior

dfba_sim_data(n = 50,
```

```

a0 = .5,
b0 = .5,
model = "cauchy",
design = "paired",
delta=.4)

```

dfba_wilcoxon

Repeated-Measures Test (Wilcoxon Signed-Ranks Test)

Description

Given two continuous, paired variates Y1 and Y2, computes the sample T_pos and T_neg statistics for the Wilcoxon signed-rank test and provides a Bayesian analysis for the population sign-bias parameter phi_w, which is the population proportion of positive differences.

Usage

```

dfba_wilcoxon(
  Y1,
  Y2,
  a0 = 1,
  b0 = 1,
  prob_interval = 0.95,
  samples = 30000,
  method = NULL,
  hide_progress = FALSE
)

```

Arguments

Y1	Numeric vector for one continuous variate
Y2	Numeric vector for values paired with Y1 variate
a0	The first shape parameter for the prior beta distribution for phi_w. Must be positive and finite.
b0	The second shape parameter for the prior beta distribution for phi_w. Must be positive and finite.
prob_interval	Desired probability for interval estimates of the sign bias parameter phi_w (default is 0.95)
samples	When method = "small", the number of desired Monte Carlo samples per candidate value for phi_w (default is 30000 per candidate phi)
method	(Optional) The method option is either "small" or "large". The "small" algorithm is based on a discrete Monte Carlo solution for cases where n is typically less than 20. The "large" algorithm is based on beta approximation model for the posterior distribution for the phi_w parameter. This approximation is reasonable when $n > 19$. Regardless of n the user can stipulate either method. When the method argument is omitted, the program selects the appropriate procedure.
hide_progress	(Optional) If TRUE, hide percent progress while Monte Carlo sampling is running when method = SMALL. (default is FALSE).

Details

The Wilcoxon signed-rank test is the frequentist nonparametric counterpart to the paired t -test. The procedure is based on the rank of the difference scores $d = Y1 - Y2$. The ranking is initially done on the absolute value of the nonzero d values, and each rank is then multiplied by the sign of the difference. Differences equal to zero are dropped. Since the procedure is based on only ranks of the differences, it is robust with respect to outliers in either the $Y1$ or $Y2$ measures. The procedure does not depend on the assumption of a normal distribution for the two continuous variates.

The sample T_{pos} statistic is the sum of the ranks that have a positive sign, whereas T_{neg} is the positive sum of the ranks that have a negative value. Given n nonzero d scores, $T_{\text{pos}} + T_{\text{neg}} = n(n + 1)/2$. Tied ranks are possible, especially when there are $Y1$ and $Y2$ values that have low precision. In such cases, the Wilcoxon statistics are rounded to the nearest integer.

The Bayesian analysis is based on a parameter ϕ_w , which is the population proportion for positive d scores. The default prior for ϕ_w is a flat beta distribution with shape parameters $a_0 = b_0 = 1$, but the user can stipulate their preferred beta prior by assigning values for a_0 and b_0 . The `prob_interval` input, which has a default value of .95, is the value for interval estimates for the ϕ_w parameter, but the user can alter this value if they prefer.

There are two cases for the Bayesian analysis - one for a small number of pairs and another for when there is a large number of pairs. The `method = "small"` sample algorithm uses a discrete approximation where there are 200 candidate values for ϕ_w , which are .0025 to .9975 in steps of .005. For each candidate value for ϕ_w , there is a prior and posterior probability. The posterior probability is based on Monte Carlo sampling to approximate the likelihood for obtaining the observed Wilcoxon statistics. That is, for each candidate value for ϕ_w , thousands of Monte Carlo samples are generated for the signs on the numbers (1,2, ..., n) where each number is multiplied by the sign. The proportion of the samples that result in the observed Wilcoxon statistics is an estimate for the likelihood value for that candidate ϕ_w . The likelihood values along with the prior result in a discrete posterior distribution for ϕ_w . The default for the number of Monte Carlo samples per candidate ϕ_w is the input quantity called `samples`. The default value for `samples` is 30000, but this quantity can be altered by the user.

Chechile (2018) empirically found that for large n there was a beta distribution that approximated the quantiles of the discrete, small sample approach. This approximation is reasonably accurate for $n > 24$, and is used when `method = "large"`.

If the `method` argument is omitted, the function employs the method that is appropriate given the sample size. Note: the `method = "small"` algorithm is slower than the algorithm for `method = "large"`; for cases where $n > 24$, `method = "small"` and `method = "large"` will produce similar estimates but the former method requires increased processing time.

Value

A list containing the following components:

<code>T_pos</code>	Sum of the positive ranks in the pairwise comparisons
<code>T_neg</code>	Sum of the negative ranks in the pairwise comparisons
<code>n</code>	Number of nonzero differences for differences $d = Y1 - Y2$
<code>prob_interval</code>	User-defined probability for interval estimates for ϕ_w
<code>samples</code>	The number of Monte Carlo samples per candidate ϕ_w for <code>method = "small"</code> (default is 30000)
<code>method</code>	A character string that is either <code>"small"</code> or <code>"large"</code> for the algorithm used (default is <code>NULL</code>)
<code>a0</code>	The first shape parameter for the beta prior distribution (default is 1)

b0	The second shape parameter for the beta distribution prior (default is 1)
a_post	First shape parameter for the posterior beta distribution
b_post	Second shape parameter for the posterior beta distribution
phiv	The 200 candidate values for phi_w for method = "small"
phipost	The discrete posterior distribution for phi_w when method = "small"
priorprH1	The prior probability that phi_w > .5
prH1	The posterior probability for phi_w > .5
BF10	Bayes factor for the relative increase in the posterior odds for the alternative hypothesis that phi_w > .5 over the null model for phi_w <= .5
post_mean	The posterior mean for phi_w
cumulative_phi	The posterior cumulative distribution for phi_w when method = "small"
hdi_lower	The lower limit for the posterior highest-density interval estimate for phi_w
hdi_upper	The upper limit for the posterior highest-density interval estimate for phi_w
a_post	The first shape parameter for a beta distribution model for phi_w when method = "large"
b_post	The second shape parameter for a beta distribution model for phi_w when method = "large"
post_median	The posterior median for phi_w when method = "large"
eti_lower	The equal-tail lower limit for phi_w
eti_upper	The equal-tail upper limit for phi_w

References

Chechile, R.A. (2020). Bayesian Statistics for Experimental Scientists: A General Introduction to Distribution-Free Methods. Cambridge: MIT Press.

Chechile, R. A. (2018) A Bayesian analysis for the Wilcoxon signed-rank statistic. Communications in Statistics - Theory and Methods, <https://doi.org/10.1080/03610926.2017.1388402>

Examples

```
# Note: examples with method = "small" have long runtimes due to Monte Carlo
# sampling; please feel free to run them in the console.
```

```
## Examples with a small number of pairs
```

```
conditionA <- c(1.49, 0.64, 0.96, 2.34, 0.78, 1.29, 0.72, 1.52, 0.62, 1.67,
               1.19, 0.86)
conditionB <- c(0.53, 0.55, 0.58, 0.97, 0.60, 0.22, 0.05, 13.14, 0.63, 0.33,
               0.91, 0.37)
```

```
## Not run:
dfba_wilcoxon(Y1 = conditionA,
              Y2 = conditionB,
              hide_progress = FALSE)
```

```
## End(Not run)
```

```

# Note the results for this method="small" analysis differs from
# the previously run. These differences are the differences from
# different Monte Carlo sampling

# Using the Jeffreys prior for the same two conditions.

## Not run:
dfba_wilcoxon(conditionA,
               conditionB,
               a0 = .5,
               b0 = .5,
               hide_progress = FALSE)

## End(Not run)

# Using 99% interval estimates and with 50000 Monte Carlo samples per
# candidate phi_w

## Not run:

dfba_wilcoxon(conditionA,
               conditionB,
               prob_interval=.99,
               samples=50000,
               hide_progress = FALSE)

## End(Not run)

# Examples with large sample size

E <- c(6.45, 5.65, 4.34, 5.92, 2.84, 13.06, 6.61, 5.47, 4.49, 6.39, 6.63,
       3.55, 3.76, 5.61, 7.45, 6.41, 10.16, 6.26, 8.46, 2.29, 3.16, 5.68,
       4.13, 2.94, 4.87, 4.44, 3.13, 8.87)

C <- c(2.89, 4.19, 3.22, 6.50, 3.10, 4.19, 5.13, 3.77, 2.71, 2.58, 7.59,
       2.68, 4.98, 2.35, 5.15, 8.46, 3.77, 8.83, 4.06, 2.50, 5.48, 2.80,
       8.89, 3.19, 9.36, 4.58, 2.94, 4.75)

BW<-dfba_wilcoxon(Y1 = E,
                  Y2 = C)

BW
plot(BW)

# Forcing the method="small" despite a sufficiently large n value

## Not run:
CW<-dfba_wilcoxon(Y1 = E,
                  Y2 = C,
                  method = "small",
                  hide_progress = FALSE)

CW
plot(CW)
plot(CW, plot.prior = FALSE)

## End(Not run)

```

show,dfba_beta_contrast_out-method

Formats for Beta Contrasts

Description

Formats for Beta Contrasts

Usage

```
## S4 method for signature 'dfba_beta_contrast_out'
show(object)
```

```
## S4 method for signature 'dfba_beta_contrast_out'
plot(x)
```

Arguments

object	An object of class dfba_beta_contrast_out
x	object An object of class dfba_beta_contrast_out

show,dfba_beta_descriptive_out-method

Formats for Beta Descriptive

Description

Formats for Beta Descriptive

Usage

```
## S4 method for signature 'dfba_beta_descriptive_out'
show(object)
```

```
## S4 method for signature 'dfba_beta_descriptive_out'
plot(x)
```

Arguments

object	An object of class dfba_beta_descriptive_out
x	An object of class dfba_beta_descriptive_out

show,dfba_binomial_out-method

Formats for Bayesian Binomial Test

Description

Formats for Bayesian Binomial Test

Usage

```
## S4 method for signature 'dfba_binomial_out'
show(object)
```

```
## S4 method for signature 'dfba_binomial_out'
plot(x, plot.prior = TRUE)
```

Arguments

object	An object of class <code>dfba_binomial_out</code>
x	An object of class <code>dfba_binomial_out</code>
plot.prior	Show prior distribution (default = TRUE)

show,dfba_bivariate_concordance_out-method

Formatted output for dfba_bivariate_concordance

Description

Formatted output for dfba_bivariate_concordance

Usage

```
## S4 method for signature 'dfba_bivariate_concordance_out'
show(object)
```

```
## S4 method for signature 'dfba_bivariate_concordance_star_out'
show(object)
```

```
## S4 method for signature 'dfba_bivariate_concordance_out'
plot(x, plot.prior = TRUE)
```

```
## S4 method for signature 'dfba_bivariate_concordance_star_out'
plot(x, plot.prior = TRUE)
```

Arguments

object	An object of class <code>dfba_bivariate_concordance_star_out</code>
x	An object of class <code>dfba_bivariate_concordance_star_out</code>
plot.prior	Show prior distribution (default = TRUE)

show,dfba_gamma_out-method

Formatted output for dfba_gamma

Description

Formatted output for dfba_gamma

Usage

```
## S4 method for signature 'dfba_gamma_out'  
show(object)
```

```
## S4 method for signature 'dfba_gamma_out'  
plot(x, plot.prior = TRUE)
```

Arguments

object	An object of class <code>dfba_gamma_out</code>
x	An object of class <code>dfba_gamma_out</code>
plot.prior	Show prior distribution (default = TRUE)

show,dfba_interval_BF_out-method

Formats for Interval Bayes Factor

Description

Formats for Interval Bayes Factor

Usage

```
## S4 method for signature 'dfba_interval_BF_out'  
show(object)
```

Arguments

object	An object of class <code>dfba_interval_BF_out</code>
--------	--

show,dfba_mann_whitney_large_out-method
Formats for large-n Mann Whitney

Description

Formats for large-n Mann Whitney

Usage

```
## S4 method for signature 'dfba_mann_whitney_large_out'  
show(object)  
  
## S4 method for signature 'dfba_mann_whitney_large_out'  
plot(x, plot.prior = TRUE)
```

Arguments

object	An object of class <code>dfba_mann_whitney_large_out</code>
x	An object of class <code>dfba_mann_whitney_small_out</code>
plot.prior	Show prior distribution (default = TRUE)

show,dfba_mann_whitney_small_out-method
Formats for small-n Mann Whitney

Description

Formats for small-n Mann Whitney

Usage

```
## S4 method for signature 'dfba_mann_whitney_small_out'  
show(object)  
  
## S4 method for signature 'dfba_mann_whitney_small_out'  
plot(x, plot.prior = TRUE)
```

Arguments

object	An object of class <code>dfba_mann_whitney_small_out</code>
x	An object of class <code>dfba_mann_whitney_small_out</code>
plot.prior	Show prior distribution (default = TRUE)

show,dfba_mcnemar_out-method

Format for Bayesian McNemar Test

Description

Format for Bayesian McNemar Test

Usage

```
## S4 method for signature 'dfba_mcnemar_out'
show(object)
```

```
## S4 method for signature 'dfba_mcnemar_out'
plot(x, plot.prior = TRUE)
```

Arguments

object	An object of class dfba_mcnemar_out
x	An object of class dfba_mcnemar_out
plot.prior	Show prior distribution (default = TRUE)

show,dfba_median_test_out-method

Formats for Bayesian Median Test Output

Description

Formats for Bayesian Median Test Output

Usage

```
## S4 method for signature 'dfba_median_test_out'
show(object)
```

Arguments

object	An object of class dfba_median_test_out
--------	---

show,dfba_point_BF_out-method

Formats for Point Bayes Factor

Description

Formats for Point Bayes Factor

Usage

```
## S4 method for signature 'dfba_point_BF_out'
show(object)
```

Arguments

object An object of class [dfba_point_BF_out](#)

show,dfba_power_curve_out-method

Formats for power curve

Description

Formats for power curve

Usage

```
## S4 method for signature 'dfba_power_curve_out'
show(object)
```

```
## S4 method for signature 'dfba_power_curve_out'
plot(x)
```

Arguments

object An object of class [dfba_power_curve_out](#)

x An object of class [dfba_power_curve_out](#)

show,dfba_sign_test_out-method

Formats for Bayesian Sign Test

Description

Formats for Bayesian Sign Test

Usage

```
## S4 method for signature 'dfba_sign_test_out'
show(object)
```

```
## S4 method for signature 'dfba_sign_test_out'
plot(x, plot.prior = TRUE)
```

Arguments

object	An object of class <code>dfba_sign_test_out</code>
x	An object of class <code>dfba_sign_test_out</code>
plot.prior	Show prior distribution (default = TRUE)

show,dfba_sim_data_out-method

Format for Simulated Data Function

Description

Format for Simulated Data Function

Usage

```
## S4 method for signature 'dfba_sim_data_out'
show(object)
```

```
## S4 method for signature 'dfba_sim_data_out'
plot(x)
```

Arguments

object	An object of class <code>dfba_sim_data_out</code>
x	An object of class <code>dfba_sim_data_out</code>

show,dfba_t_power_out-method

Bayesian vs. t Power Methods

Description

Bayesian vs. t Power Methods

Usage

```
## S4 method for signature 'dfba_t_power_out'  
show(object)
```

```
## S4 method for signature 'dfba_t_power_out'  
plot(x)
```

Arguments

object	An object of class <code>dfba_t_power_out</code>
x	An object of class <code>dfba_t_power_out</code>

show,dfba_wilcoxon_large_out-method

Formats for large-n Wilcoxon

Description

Formats for large-n Wilcoxon

Usage

```
## S4 method for signature 'dfba_wilcoxon_large_out'  
show(object)
```

```
## S4 method for signature 'dfba_wilcoxon_large_out'  
plot(x, plot.prior = TRUE)
```

Arguments

object	An object of class <code>dfba_wilcoxon_large_out</code>
x	An object of class <code>dfba_wilcoxon_large_out</code>
plot.prior	Show prior distribution (default = TRUE)

`show,dfba_wilcoxon_small_out-method`*Formats for small-n Wilcoxon*

Description

Formats for small-n Wilcoxon

Usage

```
## S4 method for signature 'dfba_wilcoxon_small_out'  
show(object)
```

```
## S4 method for signature 'dfba_wilcoxon_small_out'  
plot(x, plot.prior = TRUE)
```

Arguments

<code>object</code>	An object of class <code>dfba_wilcoxon_small_out</code>
<code>x</code>	An object of class <code>dfba_wilcoxon_small_out</code>
<code>plot.prior</code>	Show prior distribution (default = TRUE)

Index

* internal

- DFBA-package, [2](#)
- show,dfba_beta_contrast_out-method, [49](#)
- show,dfba_beta_descriptive_out-method, [49](#)
- show,dfba_binomial_out-method, [50](#)
- show,dfba_bivariate_concordance_out-method, [50](#)
- show,dfba_gamma_out-method, [51](#)
- show,dfba_interval_BF_out-method, [51](#)
- show,dfba_mann_whitney_large_out-method, [52](#)
- show,dfba_mann_whitney_small_out-method, [52](#)
- show,dfba_mcnemar_out-method, [53](#)
- show,dfba_median_test_out-method, [53](#)
- show,dfba_point_BF_out-method, [54](#)
- show,dfba_power_curve_out-method, [54](#)
- show,dfba_sign_test_out-method, [55](#)
- show,dfba_sim_data_out-method, [55](#)
- show,dfba_t_power_out-method, [56](#)
- show,dfba_wilcoxon_large_out-method, [56](#)
- show,dfba_wilcoxon_small_out-method, [57](#)
- DFBA (DFBA-package), [2](#)
- DFBA-package, [2](#)
- dfba_bayes_vs_t_power, [2](#)
- dfba_beta_bayes_factor, [7](#), [18](#), [30](#), [33](#), [40](#)
- dfba_beta_contrast, [10](#)
- dfba_beta_contrast_out, [49](#)
- dfba_beta_contrast_out-class (dfba_gamma_out-class), [24](#)
- dfba_beta_descriptive, [13](#), [18](#), [40](#)
- dfba_beta_descriptive_out, [49](#)
- dfba_beta_descriptive_out-class (dfba_gamma_out-class), [24](#)
- dfba_binomial, [15](#)
- dfba_binomial_out, [50](#)
- dfba_binomial_out-class (dfba_gamma_out-class), [24](#)
- dfba_bivariate_concordance, [19](#), [24](#)
- dfba_bivariate_concordance_out-class (dfba_gamma_out-class), [24](#)
- dfba_bivariate_concordance_star_out, [50](#)
- dfba_bivariate_concordance_star_out-class (dfba_gamma_out-class), [24](#)
- dfba_gamma, [22](#)
- dfba_gamma_out, [51](#)
- dfba_gamma_out-class, [24](#)
- dfba_interval_BF_out, [51](#)
- dfba_interval_BF_out-class (dfba_gamma_out-class), [24](#)
- dfba_mann_whitney, [6](#), [24](#), [33](#), [37](#), [44](#)
- dfba_mann_whitney_large_out, [52](#)
- dfba_mann_whitney_large_out-class (dfba_gamma_out-class), [24](#)
- dfba_mann_whitney_small_out, [52](#)
- dfba_mann_whitney_small_out-class (dfba_gamma_out-class), [24](#)
- dfba_mcnemar, [29](#)
- dfba_mcnemar_out, [53](#)
- dfba_mcnemar_out-class (dfba_gamma_out-class), [24](#)
- dfba_median_test, [31](#)
- dfba_median_test_out, [53](#)
- dfba_median_test_out-class (dfba_gamma_out-class), [24](#)
- dfba_point_BF_out, [54](#)
- dfba_point_BF_out-class (dfba_gamma_out-class), [24](#)
- dfba_power_curve, [34](#)
- dfba_power_curve_out, [54](#)
- dfba_power_curve_out-class (dfba_gamma_out-class), [24](#)
- dfba_sign_test, [38](#)
- dfba_sign_test_out, [55](#)
- dfba_sign_test_out-class (dfba_gamma_out-class), [24](#)
- dfba_sim_data, [6](#), [37](#), [41](#)
- dfba_sim_data_out, [55](#)

- dfba_sim_data_out-class
 - (dfba_gamma_out-class), [24](#)
- dfba_t_power_out, [56](#)
- dfba_t_power_out-class
 - (dfba_gamma_out-class), [24](#)
- dfba_wilcoxon, [6](#), [37](#), [40](#), [44](#), [45](#)
- dfba_wilcoxon_large_out, [56](#)
- dfba_wilcoxon_large_out-class
 - (dfba_gamma_out-class), [24](#)
- dfba_wilcoxon_small_out, [57](#)
- dfba_wilcoxon_small_out-class
 - (dfba_gamma_out-class), [24](#)
- Distributions, [6](#), [15](#), [18](#), [37](#), [44](#)
- plot, dfba_beta_contrast_out-method
 - (show, dfba_beta_contrast_out-method), [49](#)
- plot, dfba_beta_descriptive_out-method
 - (show, dfba_beta_descriptive_out-method), [49](#)
- plot, dfba_binomial_out-method
 - (show, dfba_binomial_out-method), [50](#)
- plot, dfba_bivariate_concordance_out-method
 - (show, dfba_bivariate_concordance_out-method), [50](#)
- plot, dfba_bivariate_concordance_star_out-method
 - (show, dfba_bivariate_concordance_out-method), [50](#)
- plot, dfba_gamma_out-method
 - (show, dfba_gamma_out-method), [51](#)
- plot, dfba_mann_whitney_large_out-method
 - (show, dfba_mann_whitney_large_out-method), [52](#)
- plot, dfba_mann_whitney_small_out-method
 - (show, dfba_mann_whitney_small_out-method), [52](#)
- plot, dfba_mcnemar_out-method
 - (show, dfba_mcnemar_out-method), [53](#)
- plot, dfba_power_curve_out-method
 - (show, dfba_power_curve_out-method), [54](#)
- plot, dfba_sign_test_out-method
 - (show, dfba_sign_test_out-method), [55](#)
- plot, dfba_sim_data_out-method
 - (show, dfba_sim_data_out-method), [55](#)
- plot, dfba_t_power_out-method
 - (show, dfba_t_power_out-method), [56](#)
- plot, dfba_wilcoxon_large_out-method
 - (show, dfba_wilcoxon_large_out-method), [56](#)
- plot, dfba_wilcoxon_small_out-method
 - (show, dfba_wilcoxon_small_out-method), [57](#)
- show, dfba_beta_contrast_out-method, [49](#)
- show, dfba_beta_descriptive_out-method, [49](#)
- show, dfba_binomial_out-method, [50](#)
- show, dfba_bivariate_concordance_out-method, [50](#)
- show, dfba_bivariate_concordance_star_out-method
 - (show, dfba_bivariate_concordance_out-method), [50](#)
- show, dfba_gamma_out-method, [51](#)
- show, dfba_interval_BF_out-method, [51](#)
- show, dfba_mann_whitney_large_out-method, [52](#)
- show, dfba_mann_whitney_small_out-method, [52](#)
- show, dfba_mcnemar_out-method, [53](#)
- show, dfba_median_test_out-method, [53](#)
- show, dfba_point_BF_out-method, [54](#)
- show, dfba_power_curve_out-method, [54](#)
- show, dfba_sign_test_out-method, [55](#)
- show, dfba_sim_data_out-method, [55](#)
- show, dfba_t_power_out-method, [56](#)
- show, dfba_wilcoxon_large_out-method, [56](#)
- show, dfba_wilcoxon_small_out-method, [57](#)